Characterizing and Measuring Linguistic Dataset Drift

Tyler A. Chang[1*] **Kishaloy Halder**[2†] **Neha Anna John**[2] **Yogarshi Vyas**[2]
Yassine Benajiba[2] **Miguel Ballesteros**[2] **Dan Roth**[2]
[1]University of California San Diego
[2]AWS AI Labs
tachang@ucsd.edu
{kishaloh,nehajohn,yogarshi,benajiy,ballemig,drot}@amazon.com

Abstract

NLP models often degrade in performance when real world data distributions differ markedly from training data. However, existing dataset drift metrics in NLP have generally not considered specific dimensions of linguistic drift that affect model performance, and they have not been validated in their ability to predict model performance at the individual example level, where such metrics are often used in practice. In this paper, we propose three dimensions of linguistic dataset drift: vocabulary, structural, and semantic drift. These dimensions correspond to content word frequency divergences, syntactic divergences, and meaning changes not captured by word frequencies (e.g. lexical semantic change). We propose interpretable metrics for all three drift dimensions, and we modify past performance prediction methods to predict model performance at both the example and dataset level for English sentiment classification and natural language inference. We find that our drift metrics are more effective than previous metrics at predicting out-of-domain model accuracies (mean 16.8% root mean square error decrease), particularly when compared to popular fine-tuned embedding distances (mean 47.7% error decrease). Fine-tuned embedding distances are much more effective at ranking individual examples by expected performance, but decomposing into vocabulary, structural, and semantic drift produces the best example rankings of all considered model-agnostic drift metrics (mean 6.7% ROC AUC increase).

1 Introduction

Dataset drift, when test data distributions differ from a model's training data, can have detrimental effects on NLP model performance (Broscheit et al., 2022; Do et al., 2021; Koh et al., 2021). In real world scenarios, models are regularly monitored for potential performance degradations by comparing incoming test data with the training data (Elango et al., 2022; Nigenda et al., 2022). For these scenarios, researchers have proposed a variety of linguistic dataset drift metrics that aim to predict NLP model performance degradations between training and test domains (Elsahar and Gallé, 2019; Ramesh Kashyap et al., 2021).

However, previous drift metrics and performance predictions suffer from several limitations. First, previous metrics have generally been designed as holistic measures of linguistic dataset drift, despite the fact that different NLP tasks and models might be sensitive to different dimensions of linguistic drift. Second, previous research has focused on drift metrics at the dataset level rather than the individual example level. Not only does this require multiple labeled evaluation domain datasets to make out-of-domain performance predictions (regressions require multiple dataset-level drift values to fit to; Elsahar and Gallé, 2019; Ramesh Kashyap et al., 2021), but drift metrics are often used in practice to predict model performance when individual real world examples are streamed in real time (Elango et al., 2022). We seek to overcome both of these limitations by proposing and evaluating specific dimensions of linguistic drift, predicting out-of-domain model performance at both the individual example level and the dataset level.

Specifically, we propose three dimensions of linguistic dataset drift along with corresponding drift metrics: vocabulary, structural, and semantic drift. Because these dimensions capture distinct features that could have different effects on the performance of an NLP model, we hypothesize that decomposing into these three dimensions will allow NLP performance prediction models to better predict model performance on novel data. Indeed, when compared to previous model-agnostic drift metrics predicting performance on English sentiment classification and natural language inference (NLI), our metrics produce both improved predictions

[*] Work done during an internship at AWS AI Labs.
[†] Corresponding author.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 8953–8967
July 9-14, 2023 ©2023 Association for Computational Linguistics

of dataset-level accuracies and improved rankings of individual examples by expected performance, for both in-domain and out-of-domain data (mean 16.8% accuracy root mean square error decrease, mean 6.7% ROC area under the curve increase). Although we find that previously-proposed fine-tuned embedding distances (Elango et al., 2022) are far more effective at ranking individual examples by expected performance, those distances are extremely ineffective at predicting actual model accuracies. We conclude that decomposing linguistic drift into vocabulary, structural, and semantic drift is an effective approach for predicting out-of-domain model accuracy, and for ranking individual examples when model-agnostic metrics are desired.

2 Related Work

Past work has quantified the drift between NLP datasets using distances between token frequency distributions or TF-IDF vectors (Bäck, 2019; Ramesh Kashyap et al., 2021; Sato et al., 2022), language model embedding distances (Feldhans et al., 2021; Yamshchikov et al., 2021), or the ability of domain classifiers to discriminate between datasets (Dredze et al., 2010; Elsahar and Gallé, 2019; Ruder et al., 2017). Notably, Ramesh Kashyap et al. (2021) find that these metrics can predict performance degradations when an NLP model is transferred from a training dataset D_{train} to an out-of-domain evaluation dataset D_{eval}.

However, existing metrics have generally been designed as holistic measures of linguistic drift, failing to capture specific dimensions that might affect NLP model performance in different ways. Furthermore, the traditional setup for evaluating drift metrics (Elsahar and Gallé, 2019; Ramesh Kashyap et al., 2021) only allows for dataset-level drift metrics that predict overall model accuracy on out-of-domain datasets. In practice, when real world examples are streamed during test time, it is desirable to predict model performance for individual examples using example-level drift metrics (i.e. drift between an example x and a training dataset D_{train}; Elango et al., 2022; Nigenda et al., 2022). In our work, we modify the setup from Ramesh Kashyap et al. (2021) to predict performance for individual examples (Section 4), using logistic regressions fitted to example-level drift metrics. In contrast to Ramesh Kashyap et al. (2021), we can fit our regressions to predict out-of-domain performance even when only a single in-domain evaluation dataset is available.

3 Dimensions of Linguistic Drift

As described above, previous measures of dataset drift in NLP suffer from (1) lack of specificity and (2) lack of validation at the example level, where such metrics are often used in practice. First, we address the lack of specificity by proposing three dimensions of linguistic dataset drift: vocabulary, structural, and semantic drift. As in previous work, we primarily focus on *domain* drift, *i.e.* divergence in the input probabilities $P(x)$ rather than the joint probabilities over inputs and labels $P(x, y)$. For each of our proposed drift dimensions, we propose a metric that quantifies the drift between an evaluation example x and a training dataset D_{train}, allowing us to use our metrics to predict example-level model performance. We evaluate our metrics empirically in Section 4.

3.1 Vocabulary Drift

We define vocabulary drift as the divergence between content word frequencies in two text samples. Content words are defined as open class words that generally contain substantial semantic content (e.g. nouns, verbs, adjectives, and adverbs), contrasted with function words that primarily convey grammatical relationships (e.g. prepositions, conjunctions, and pronouns; Bell et al., 2009; Segalowitz and Lane, 2000). By restricting our vocabulary drift definition to content word distributions, we capture vocabulary differences between two text datasets without the confounds of structural features. For example, *"The student ate the sandwich"* and *"A sandwich was eaten by a student"* would have low vocabulary drift after excluding function words. Notably, our definition of vocabulary drift is designed to include drift in word choice, regardless of the semantic similarity between chosen words; for example, *"The dog was happy"* and *"The beagle was ecstatic"* would have high vocabulary drift due to differing word choice, despite their high semantic similarity. This property is useful because NLP models are often sensitive to changes in word choice even if datasets are semantically similar (Hu et al., 2019; Misra et al., 2020).

Formally, to quantify the vocabulary drift between an evaluation example x and a training dataset D_{train}, we compute the cross-entropy between content word frequencies in x and D_{train} as:

$$\frac{1}{|x_{\text{content}}|} \sum_{w \in x_{\text{content}}} \log(P_{\text{train\_content}}(w)). \quad (1)$$

Here, x_{content} is the set of content words in example x, and $P_{\text{train\_content}}(w)$ is the frequency (restricted to content words) of word w in the training dataset. Our vocabulary drift metric is equal to the log-perplexity (training loss) of a unigram language model restricted to content words, trained on D_{train} and evaluated on x. We annotate content words using the spaCy tokenizer and part-of-speech (POS) tagger (Honnibal et al., 2017), defining content words as those with an open class Universal POS tag (nouns, verbs, adjectives, adverbs, and interjections; Nivre et al., 2020) and excluding stop words in spaCy.

3.2 Structural Drift

In contrast to vocabulary drift, structural drift captures divergences between the syntactic structures in two text samples. For example, *"Yesterday, I was surprised by a dog"* and *"Usually, she is recognized by the audience"* would have low structural drift despite high vocabulary drift. Previous work in discourse analysis has attempted to quantify structural similarity separately from semantic similarity in natural conversations, although their metrics are not directly applicable to NLP datasets due to computational limitations (Boghrati et al., 2018).[1] Structural drift has also been studied in machine translation, primarily considering structural divergence between parallel text in different languages (Dave et al., 2004; Deng and Xue, 2017; Dorr, 1990; Saboor and Khan, 2010); in our work, we focus on divergences between non-parallel monolingual text.

We quantify the structural drift between an example x and D_{train} using the cross-entropy between the true POS tag sequence for x and the predictions of a POS 5-gram model trained on POS tag sequences in D_{train}. This metric captures the divergence between syntactic structures in x and D_{train} using 5-gram sequences, abstracting away from semantic content and vocabulary by considering only the POS tag for each word (Axelrod et al., 2015; Nerbonne and Wiersma, 2006). Formally, we compute:

$$\frac{1}{|x|} \sum_{i=1}^{|x|} \log(P_{\text{train}}(\text{tag}_i|\text{tag}_{i-1}, ..., \text{tag}_{i-4})). \quad (2)$$

We pad the beginning of the POS tag sequence with [SEP] tokens, and we only annotate examples with structural drift if they contain at least two non-[SEP] tokens. As with our vocabulary drift metric, we annotate POS tags using the spaCy tokenizer and POS tagger.

3.3 Semantic Drift

Finally, we consider semantic drift, defined as any divergence in semantic meaning between two text samples. Semantic drift is closely related to both vocabulary and structural drift; the words and syntactic structures used in a sentence are closely tied to the meaning of that sentence, particularly under compositional assumptions of language (Szabó, 2022). However, there are notable cases where semantic drift is independent from vocabulary and structural drift. For example, *"I saw the doctor"* and *"I took a trip to the hospital"* have high vocabulary and structural drift under our definitions, despite similar semantic meaning. Conversely, some sentences have different meanings or connotations across time and contexts, despite remaining identical in both vocabulary and structure (e.g. the word *"sick"* in *"That salamander is sick!"* can mean very cool or physically ill depending on the context).

Many of these semantic similarities and differences can be quantified using contextualized embeddings from modern language models (Briakou and Carpuat, 2020; Devlin et al., 2019; Liu et al., 2020; Sun et al., 2022), which we include in our drift metric experiments (Section 4). However, when identifying individual dimensions of linguistic drift, we seek to identify dimensions that are both interpretable and relatively independent from one another, to better isolate specific dimensions that impact NLP model performance. Language model embeddings reflect vocabulary and structural properties of sentences as well as semantic properties (Hewitt and Manning, 2019; Tenney et al., 2019), and thus they are less effective for pinpointing interpretable effects that are specific to semantic drift.

Lexical Semantic Change. Instead, we consider lexical semantic change, in which a word's meaning changes between two datasets while its sur-

[1] The CASSIM structural similarity metric in Boghrati et al. (2018) is based on tree-edit distances between all sentence pairs, which is slow to compute even for relatively small NLP datasets.

face form remains the same (Gulordava and Baroni, 2011; Kulkarni et al., 2015; Sagi et al., 2009; Tahmasebi et al., 2021). Past work has quantified a token's lexical semantic change $\text{LSC}_{D_1 \leftrightarrow D_2}(w)$ using the mean pairwise cosine distance between contextualized RoBERTa embeddings for that token in two different datasets D_1 and D_2 (Giulianelli et al., 2020; Laicher et al., 2021). Motivated by this metric, we quantify the lexical semantic change between an evaluation example x and a training dataset D_{train} using the mean lexical semantic change between x and D_{train} for all content tokens w shared between x and D_{train}:

$$\frac{1}{|x_{\text{content}}|} \sum_{w \in x_{\text{content}}} \text{LSC}_{x \leftrightarrow D_{\text{train}}}(w). \tag{3}$$

Here, $\text{LSC}_{x \leftrightarrow D_{\text{train}}}(w)$ is the mean pairwise cosine distance between embeddings for w in example x and dataset D_{train}, using a non-fine-tuned RoBERTa model. Again, we define content tokens as tokens that are annotated with an open class POS tag anywhere in the Universal Dependencies English dataset, excluding stop words (Nivre et al., 2020).[2] While this lexical semantic change metric is still based on contextualized embeddings, matching embeddings based on token surface forms allows us to minimize effects of vocabulary and structural drift, as compared to matching each example representation with all other example representations regardless of surface form. Of course, lexical semantic change is just one type of semantic drift; future work might consider other types of semantic drift that are independent from vocabulary and structural drift.

4 Experiments

Previous work has evaluated drift metrics by assessing their ability to predict out-of-domain model performance at the dataset-level using dataset-level metrics (e.g. Ramesh Kashyap et al., 2021; Section 2). We extend this work by predicting individual example-level performance (probabilities of getting individual examples correct) along with dataset-level accuracies, using drift metrics between each example x and the training dataset D_{train}. Using these example-level metrics instead of dataset-level metrics allows us to fit regressions predicting model performance using only a set of examples (e.g. using only the in-domain evaluation set), rather than a set of multiple evaluation datasets covering different domains. Thus, our approach can be used in common real world scenarios where labeled data is available only in one domain. In our experiments, we compare previous drift metrics with our proposed metrics for vocabulary, structural, and semantic drift, evaluating whether decomposing linguistic drift into these three dimensions improves NLP model performance predictions. [3]

4.1 Datasets

We evaluate cross-domain transfer performance for language models fine-tuned on sentiment classification (split by product category or review year) and natural language inference (NLI, split by source domain). Because these tasks output one prediction per sequence, they allow us to directly evaluate sequence-level (i.e. example-level) drift metrics.

Amazon Reviews (product categories). For sentiment classification, we consider the Amazon reviews dataset, containing customer-written product reviews for 43 different product categories (Amazon, 2017). As in Blitzer et al. (2007), we label 1- and 2-star reviews as negative, and 4- and 5-star reviews as positive. We sample up to 100K polarity-balanced reviews from each product category, considering each category as a domain. For each product category, we use a 70/20/10% split for training, evaluation and test datasets.

Amazon Reviews (temporal split). Next, we consider the same Amazon reviews dataset for sentiment classification, but we define domains by review date rather than by product category. We generate a category-balanced and polarity-balanced sample for each year between 2001 and 2015 (inclusive) by sampling up to 5K polarity-balanced reviews from each product category for each year, sampling the same number of reviews each year for any given category. The resulting dataset has 33K training examples, 5K evaluation examples, and 5K test examples for each year, similar to Agarwal and Nenkova (2022), but balanced for product category and polarity.

[2] We exclude non-content tokens for lexical semantic change because non-content token embeddings (e.g. for function words and punctuation) are more likely to encode structural drift information rather than lexical semantic change. Contextualized token embeddings are computed as the mean of the token representations in the last two RoBERTa layers before fine-tuning (Elango et al., 2022).

[3] Code is available at https://github.com/amazon-science/characterizing-measuring-linguistic-drift.

MultiNLI. Finally, we consider the MNLI dataset for natural language inference (NLI), covering five training domains and ten evaluation domains, including government documents, pop culture articles, and transcribed telephone conversations (Williams et al., 2018). Each training domain has approximately 77K training examples, and each evaluation domain has approximately 2K evaluation examples.

4.2 Models

We fine-tune a RoBERTa base-size model $\mathcal{M}$ for each training domain for each task, using batch size 32, learning rate 2e-5, and four epochs through the training data (Liu et al., 2019). Because there are only five training domains for MNLI, we run five fine-tuning runs per MNLI training domain. Full fine-tuning details and hyperparameters are listed in Appendix A.1. We evaluate each model on each evaluation domain; to simulate realistic scenarios for temporal data, we evaluate only on future years for models trained on temporal splits.

4.3 Drift metrics

We consider drift metrics between individual evaluation examples x and training datasets D_{train}. First, we consider our vocabulary, structural, and semantic drift metrics from Section 3. Initial motivations and theoretical examples of how these three dimensions differ are described in Section 3, but the dimensions are not perfectly independent. Empirically, Pearson correlations between our vocabulary, structural, and semantic drift metrics range from 0.10 to 0.50 across the different tasks. For comparison, we also consider drift metrics from past work: token frequency divergences and embedding cosine distances. With the exception of the fine-tuned embedding distances, all of our metrics are model-agnostic, meaning they are not dependent on the internals of the fine-tuned model.

Token frequency divergences. We compute the Jensen-Shannon (JS) divergence between the token frequency distribution for each example x and each training dataset D_{train}. This divergence has been shown to correlate with out-of-domain model performance when computed at the dataset-level (i.e. between an entire evaluation set D_{eval} and the training set D_{train}; Ramesh Kashyap et al., 2021), and it has been recommended as a metric for training dataset selection (Ruder et al., 2017).

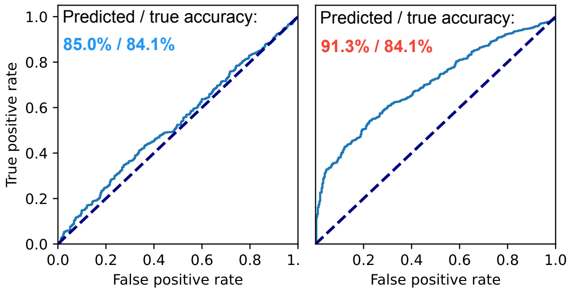

Figure 1: ROC curves predicting whether a model will get individual examples correct based on vocabulary, structural, and semantic drift (left) vs. fine-tuned embedding distances (right), for an MNLI model transferred from the telephone to fiction domain. The true model accuracy degradation is $87.9\% \rightarrow 84.1\%$. Here, our decomposed drift metrics produce worse example rankings than the fine-tuned embedding distances (ROC AUC 0.532 vs. 0.702), but a much better dataset-level accuracy prediction (absolute error 0.9% vs. 7.2%). We observe this pattern to hold across domains and datasets (Table 1). Still, our decomposed metrics outperform previous model-agnostic metrics by all evaluation criteria.

However, because example-level token frequency distributions are quite sparse (Ruder et al., 2017), we also consider the cross-entropy between each example frequency distribution and each training frequency distribution (i.e. the loss of a unigram language model). The resulting token frequency cross-entropy is equivalent to our vocabulary drift metric, but using the RoBERTa tokenizer and without the restriction to content words.

Embedding cosine distances. We compute embeddings for training and evaluation examples x by taking the mean over all tokens in x and the last two RoBERTa layers, either before or after task fine-tuning (i.e. pre-trained or fine-tuned; Elango et al., 2022). We note that the pre-trained RoBERTa model is still the same model that is fine-tuned for each task, potentially leading to overly optimistic results for the pre-trained embedding cosine distances; this caveat also holds for our semantic drift metric, which relies on pre-trained embeddings. For the embedding cosine distance drift metrics, we compute the mean cosine distance between the embedding for evaluation example x and each example in the training dataset D_{train} (Nigenda et al., 2022).[4]

[4] For efficiency, we compute mean pairwise cosine distances using the method described in Appendix A.2.

4.4 Predicting Model Performance

For each drift metric (or set of drift metrics) and each model $\mathcal{M}$ trained on dataset D_{train}, we fit a logistic regression predicting whether $\mathcal{M}$ will get example x correct (i.e. a "positive" example), based on the drift metric(s) between x and D_{train}. The regression input is the considered drift metric(s) from x to D_{train}, and the label is 1 if $\mathcal{M}$ predicts x correctly, and 0 otherwise.[5] We fit the logistic regression for all x in the in-domain evaluation dataset, mimicking a scenario where labeled evaluation data is only available in the same domain as training. This allows us to test whether regressions fitted only to in-domain examples can extrapolate to out-of-domain examples.

We evaluate the logistic regressions on both in-domain and out-of-domain evaluation examples. Each regression produces a predicted probability of "positive" (getting an example correct) for each example.[6] For dataset-level accuracy predictions, we compute the mean predicted "positive" probability over all examples in each evaluation dataset D_{eval}, equal to the expected value of model accuracy on D_{eval} based on the example-level probabilities.

4.5 Evaluating Performance Predictions

We use ROC curves to evaluate example-level performance predictions, both in-domain and out-of-domain, and we use root mean square errors (RMSEs) to evaluate out-of-domain dataset-level accuracy predictions.

ROC AUC. For each logistic regression, predicting positive examples (correct model predictions) from a given drift metric and for a given model, we compute the area under the ROC curve for in-domain and out-of-domain examples. An ROC curve plots recall (proportion of true positives identified) over the false positive rate for different probability thresholds. In our case, a higher ROC AUC indicates that the input drift metric can generally predict more true positives (examples the model gets correct) for a given false positive rate. However, ROC curves are dependent only on the rankings of examples by predicted positive probabilities (Tang et al., 2010); the raw probabilities of correct model predictions do not affect the ROC AUC as long as the example ranking is preserved. From this perspective, a higher ROC AUC indicates that evaluation examples are ranked roughly in order of expected performance; examples with higher predicted probabilities are more likely to be predicted correctly by the model. For each drift metric, we compute the mean ROC AUC over all trained models $\mathcal{M}$, for in-domain and out-of-domain examples.

RMSE. Because ROC AUCs depend only on the ranking of evaluation examples, they do not capture whether the predicted positive probabilities (probabilities of correct predictions) are actually reflective of model accuracies. For example, a given drift metric can achieve a high ROC AUC by ranking evaluation examples accurately, even if the mean probability (expected model accuracy) is far from the true model accuracy for D_{eval} (e.g. Figure 1).

Thus, for each drift metric, we also compute the RMSE comparing expected model accuracy (mean positive probability over all examples in D_{eval}) to actual model accuracy on D_{eval}. We compute RMSEs over all models $\mathcal{M}$ and their corresponding out-of-domain datasets D_{eval}. We report RMSEs as percentages of a baseline RMSE that predicts out-of-domain accuracy on D_{eval} to be the same as the in-domain evaluation accuracy (i.e. predicting no out-of-domain performance drop). Our reported RMSE percentages indicate the percentage of accuracy prediction error that remains when using a given drift metric, relative to the baseline.

To summarize, we compute the predicted accuracy RMSE and the mean ROC AUC for each drift metric and for each task. ROC AUC measures how well a drift metric ranks the evaluation examples (examples with higher "positive" probabilities should be more likely to be predicted correctly by the model), while RMSE measures how well the drift metric predicts actual model accuracy (mean probabilities should be close to the true model accuracy). An ideal drift metric should have high ROC AUC and low RMSE.

5 Results

The mean accuracy change (±standard deviation; in raw accuracy percentage difference) from in-domain to out-of-domain evaluation is $-1.04 \pm 1.02\%$ for sentiment classification across product categories, $-0.20 \pm 0.57\%$ for sentiment classification across years, and $-1.83 \pm 2.91\%$ for MNLI across source domains. Notably, in many cases, ac-

[5] In cases where we input multiple drift metrics into the logistic regression, we exclude interaction terms; interaction terms generally resulted in worse out-of-domain performance predictions, based on both ROC AUCs and RMSEs.

[6] For in-domain evaluation example predictions, we use 5-fold cross-validation, fitting regressions to only 80% of the in-domain evaluation dataset per fold.

	Sentiment (categories)			**Sentiment (temporal)**			**MNLI (source domains)**		
	In-domain	Out-of-domain		In-domain	Out-of-domain		In-domain	Out-of-domain	
Drift metric(s)	**ROC AUC** ↑	**ROC AUC** ↑	**RMSE %** ↓	**ROC AUC** ↑	**ROC AUC** ↑	**RMSE %** ↓	**ROC AUC** ↑	**ROC AUC** ↑	**RMSE %** ↓
Baseline (no-performance drop)	0.500	0.500	100.0%	0.500	0.500	100.0%	0.500	0.500	100.0%
Token frequency JS-div (Ramesh Kashyap et al., 2021; Ruder et al., 2017)	0.512	0.517	98.4%	0.519	0.528	106.2%	0.496	0.503	118.8%
Token frequency cross-entropy	0.540	0.551	71.4%	0.543	0.557	97.3%	0.500	0.512	96.8%
Cosine distance (pre-trained) (Ramesh Kashyap et al., 2021)	0.535	0.558	93.6%	0.534	0.559	91.8%	0.484	0.508	107.5%
Combined prev. model-agnostic	0.551	0.557	70.3%	0.554	0.562	142.1%	0.520	0.514	99.8%
Vocabulary drift	0.561	0.570	**51.8%**	0.552	0.571	105.8%	0.474	0.500	81.5%
Structural drift	0.572	0.575	91.4%	0.568	0.581	146.1%	0.516	**0.531**	80.6%
Semantic drift	0.586	0.591	58.4%	0.565	0.586	110.4%	0.516	0.521	**79.1%**
Vocabulary, structural, semantic drift	**0.597**	**0.601**	52.4%	**0.578**	**0.596**	**84.8%**	**0.525**	**0.531**	81.0%
Model-dependent:									
Cosine distance (fine-tuned) (Nigenda et al., 2022)	**0.845**	**0.822**	81.9%	**0.852**	**0.834**	236.7%	**0.699**	**0.683**	141.9%

Table 1: Mean ROC AUCs and RMSEs using different drift metrics to predict model performance, comparing our metrics (vocabulary, structural, and semantic drift) with previous metrics. ROC AUCs indicate the quality of example rankings by expected performance, and RMSEs (as percentages of the baseline error) indicate the quality of the actual accuracy predictions. Given in-domain accuracy p, the baseline predicts out-of-domain accuracy p and an equal probability p of getting any individual example correct. All metrics are model-agnostic except the fine-tuned embedding cosine distances.

curacy improves for out-of-domain evaluation (e.g. MNLI fiction → government). Results predicting out-of-domain evaluation accuracies (RMSEs) and example-level performance (ROC AUCs) from different drift metrics are reported in Table 1.

5.1 Ranking Examples (ROC AUC)

Mean ROC AUCs for different drift metrics are reported in Table 1, for both in-domain and out-of-domain evaluation examples. Recall that a higher ROC AUC indicates that higher scoring examples (as ranked by the logistic regression) are more likely to be predicted correctly by the model.

Decomposing drift improves rankings. Using vocabulary, structural, and semantic drift as input features into the logistic regressions results in higher ROC AUCs than any of the previous model-agnostic drift metrics, for all three multi-domain datasets and for both in-domain and out-of-domain examples (top section of Table 1). Across the three datasets, this decomposed drift improves ROC AUCs by an average of 0.039 for in-domain examples and 0.033 for out-of-domain examples when compared to the best model-agnostic drift metric from previous work.

To ensure that this is not simply a result of including three different metrics in the regression, we also consider the combination of all three model-agnostic metrics from previous work ("combined previous model-agnostic" in Table 1: token frequency JS-divergence, token frequency cross-entropy, and pre-trained embedding cosine distance). For all three datasets, the combination of previous metrics still results in worse ROC AUCs than the combination of vocabulary, structural, and semantic drift, for both in-domain and out-of-domain examples. This indicates that decomposing into vocabulary, structural, and semantic drift results in better rankings of individual examples by expected performance than previous model-agnostic drift metrics.

Fine-tuned embeddings lead to the best rankings. However, fine-tuned (model-dependent) embedding cosine distances result in by far the best rankings of examples by expected performance (higher ROC AUCs). Indeed, this is the recommended drift metric when examples need to be ranked relative to one another or relative to some threshold (e.g. when there is some threshold drift value to flag examples; Elango et al., 2022; Nigenda et al., 2022); our results validate this approach. Notably, the fine-tuned embedding distances produce quality rankings even for out-of-domain examples, despite work suggesting that fine-tuning affects the in-domain representation space differently from the out-of-domain representation space in language models (Merchant et al., 2020). Our results indicate that despite these differences between the in-domain and out-of-domain fine-tuned spaces, the fine-tuned embedding distances can still be used to rank both in-domain and out-of-domain examples by expected performance.

That said, fine-tuned embedding distances require access to the internal representations of a given model; model-agnostic metrics are still useful in cases where only model outputs can be observed, or when the same drift metric needs to apply to multiple models. For these use cases, our decomposed vocabulary, structural, and semantic drift metrics outperform previous model-agnostic metrics. Furthermore, as we observe in the next section, our decomposed drift metrics result in drastically better out-of-domain accuracy predictions than fine-tuned embedding distances, despite worse rankings of individual examples.

5.2 Predicting Model Accuracy (RMSE)

As described in Section 4.5 and shown in Figure 1, a given drift metric can produce quality rankings of examples even if the raw predicted accuracies are far from the true model accuracies. Thus, as reported in Table 1, we evaluate RMSEs using different drift metrics to predict model accuracies for out-of-domain evaluation datasets.[7]

Decomposed drift has the best accuracy predictions. Decomposing into vocabulary, structural, and semantic drift results in better dataset-level accuracy predictions (lower RMSEs) than any previous drift metric(s), for all three multi-domain datasets. Accuracy predictions based on individual dimensions vary (e.g. individual dimensions are sometimes better than including all three dimensions), but predicting out-of-domain accuracy from all three dimensions results in reliably low errors compared to previous metrics. Across the three datasets, our decomposed drift results in an average decrease of 16.8% in accuracy prediction error (RMSE) when compared to the best metric from previous work.

Fine-tuned embedding distances have poor accuracy predictions. The fine-tuned embedding distances result in worse out-of-domain accuracy predictions (higher RMSEs) than our decomposed vocabulary, structural, and semantic drift for all three multi-domain datasets. Notably, they have by far the worst out-of-domain accuracy predictions of any drift metric for MNLI and sentiment classification split temporally. Across all three datasets, fine-tuned embedding distances result in an average of 2.03x more error than our decomposed vocabulary, structural, and semantic drift. This contrasts with fine-tuned embedding distances' ability to rank individual examples by expected performance better than any other metric(s). This suggests that despite maintaining *relative* distances that are predictive of relative model performance for individual examples (high ROC AUCs), fine-tuning adjusts the example embeddings such that *raw* distances are not predictive of raw out-of-domain accuracies (high RMSEs). Concretely, the logistic regressions fit to fine-tuned embedding distances yield example-level probabilities that are highly predictive of *relative* model performance between out-of-domain examples, but quite far from the *actual* expected probabilities of getting each example correct. In practice, this suggests that fine-tuned embedding distances should be used in scenarios where the relative performance of evaluation examples is important (e.g. establishing drift threshold values), but they should not be used to predict actual out-of-domain model accuracies.

6 Discussion

We find that decomposing linguistic dataset drift into our proposed vocabulary, structural, and semantic drift metrics leads to improved out-of-domain dataset-level accuracy predictions for sentiment classification and NLI. Furthermore, our decomposed drift metrics produce better rankings of individual examples by expected performance than previous model-agnostic drift metrics (e.g. token frequency divergences and pre-trained embedding distances), both in-domain and out-of-domain. Although fine-tuned embedding distances produce by far the best example rankings, they also produce egregiously incorrect out-of-domain model accuracy predictions. Our results suggest that fine-tuned

[7] We only consider accuracy prediction RMSEs for out-of-domain datasets because sufficiently sized in-domain datasets have very low variation in model accuracy.

embedding distances should still be used in cases where examples need to be ranked by expected performance (e.g. relative to a cutoff value, as in Elango et al., 2022). Vocabulary, structural, and semantic drift should be used in cases where either (1) the internal states of a model are unavailable, which is increasingly common as models are accessed through external APIs, (2) the same metric values need to be applied across multiple models (i.e. model-agnostic metrics), or (3) raw model accuracy predictions are desired.

Our work also opens up future directions of research studying specific effects of linguistic dataset drift on NLP model performance. First, future work might assess whether there are systematic effects of particular drift dimensions on specific tasks or model architectures. Second, it might consider new types of linguistic drift, potentially extending beyond domain drift (drift in $P(x)$) to consider concept drift $P(y|x)$ in NLP (Webb et al., 2016). Finally, future work might investigate methods of quantifying drift in natural language generation, where the outputs y are linguistic data. Our work lays the groundwork for these future investigations.

7 Conclusion

We propose three dimensions of linguistic dataset drift—vocabulary, structural, and semantic drift—and we modify previous performance prediction methods to predict NLP model performance at the individual example level along with the dataset level. We validate existing drift metrics for particular use cases (e.g. fine-tuned embedding distances for example ranking), and we highlight complementary use cases where our decomposed drift metrics outperform previous metrics (e.g. when predicting model accuracies or when using model-agnostic metrics). Our work lays the foundation for future research into specific and interpretable dimensions of linguistic dataset drift, improving our ability to predict NLP model performance on real world data.

Limitations

Our work has several limitations. First, our experiments are limited by the multi-domain datasets available for sequence classification tasks, limiting both our task coverage (sentiment classification and NLI) and domain type coverage (product categories, temporal splits, and text source domains). Future work can evaluate our drift metrics on token classification tasks or even sequence-to-sequence tasks by predicting sequence-level performance (e.g. proportions of correct tokens, or example-level BLEU scores; Papineni et al., 2002) from our example-level drift metrics. Past work has already considered dataset-level drift metrics and performance predictions for token classification tasks such as named entity recognition (NER) and part-of-speech (POS) tagging (Ramesh Kashyap et al., 2021; Rijhwani and Preotiuc-Pietro, 2020), and example-level drift metrics have been used in machine translation for training data example selection (Axelrod et al., 2011; Wang et al., 2017). We hope that future work will evaluate example-level drift metrics in their ability to predict NLP model performance on this wider variety of tasks.

Second, we only consider simple logistic regressions to predict whether individual examples will be predicted correctly by different models. More complex classifiers (e.g. XGBoost; Chen and Guestrin, 2016) might improve performance predictions, particularly if more drift metrics are included as inputs, or if raw example features are included (e.g. sequence length; Ye et al., 2021). Our three dimensions of linguistic drift (vocabulary, structural, and semantic drift) represent just one way of decomposing linguistic dataset drift into distinct dimensions. We hope that future work will explore novel dimensions of linguistic drift, identifying new ways of integrating different drift metrics into NLP model performance predictions across tasks and domains.

References

Oshin Agarwal and Ani Nenkova. 2022. Temporal effects on pre-trained models for language processing tasks. *Transactions of the Association for Computational Linguistics (TACL)*.

Amazon. 2017. Amazon customer reviews dataset.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Amittai Axelrod, Yogarshi Vyas, Marianna Martindale, and Marine Carpuat. 2015. Class-based n-gram language difference models for data selection. In *Proceedings of the 12th International Workshop on Spoken Language Translation: Papers*, pages 180–187, Da Nang, Vietnam.

Alan Bell, Jason M. Brenier, Michelle Gregory, Cynthia Girand, and Dan Jurafsky. 2009. Predictability effects on durations of content and function words in conversational English. *Journal of Memory and Language*, 60(1):92–111.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic. Association for Computational Linguistics.

Reihane Boghrati, Joe Hoover, Kate M. Johnson, Justin Garten, and Morteza Dehghani. 2018. Conversation level syntax similarity metric. *Behavior Research Methods*, 50:1055–1073.

Eleftheria Briakou and Marine Carpuat. 2020. Detecting Fine-Grained Cross-Lingual Semantic Divergences without Supervision by Learning to Rank. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1563–1580, Online. Association for Computational Linguistics.

Samuel Broscheit, Quynh Do, and Judith Gaspers. 2022. Distributionally robust finetuning BERT for covariate drift in spoken language understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1970–1985, Dublin, Ireland. Association for Computational Linguistics.

Jesper Bäck. 2019. Domain similarity metrics for predicting transfer learning performance. Linköping University, Department of Computer and Information Science, Master's Thesis.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 785–794. Association for Computing Machinery.

Shachi Dave, Jignashu Parikh, and Pushpak Bhattacharyya. 2004. Interlingua-based English–Hindi machine translation and language divergence. *Machine Translation*, 16:251–304.

Dun Deng and Nianwen Xue. 2017. Translation divergences in Chinese–English machine translation: An empirical investigation. *Computational Linguistics*, 43(3):521–565.

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.

Quynh Ngoc Thi Do, Judith Gaspers, Daniil Sorokin, and Patrick Lehnen. 2021. Predicting temporal performance drop of deployed production spoken language understanding models. In *Interspeech 2021*.

Bonnie Dorr. 1990. Solving thematic divergences in machine translation. In *28th Annual Meeting of the Association for Computational Linguistics*, pages 127–134, Pittsburgh, Pennsylvania, USA. Association for Computational Linguistics.

Mark Dredze, Tim Oates, and Christine Piatko. 2010. We're not in Kansas anymore: Detecting domain changes in streams. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 585–595, Cambridge, MA. Association for Computational Linguistics.

Vikram Elango, Tony Chen, and Raghu Ramesha. 2022. Detect NLP data drift using custom Amazon SageMaker Model Monitor. AWS Machine Learning Blog. Accessed: 2022-09-01.

Hady Elsahar and Matthias Gallé. 2019. To annotate or not? predicting performance drop under domain shift. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2163–2173, Hong Kong, China. Association for Computational Linguistics.

Robert Feldhans, Adrian Wilke, Stefan Heindorf, Mohammad Hossein Shaker, Barbara Hammer, Axel-Cyrille Ngonga Ngomo, and Eyke Hüllermeier. 2021. Drift detection in text data with document embeddings. In *Intelligent Data Engineering and Automated Learning*, pages 107–118. Springer International Publishing.

Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. 2020. Analysing lexical semantic change with contextualised word representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3960–3973, Online. Association for Computational Linguistics.

Kristina Gulordava and Marco Baroni. 2011. A distributional similarity approach to the detection of semantic change in the Google Books ngram corpus. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 67–71, Edinburgh, UK. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2017. spaCy: Industrial-strength natural language processing in Python.

Ziniu Hu, Ting Chen, Kai-Wei Chang, and Yizhou Sun. 2019. Few-shot representation learning for out-of-vocabulary words. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4102–4112, Florence, Italy. Association for Computational Linguistics.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. 2021. WILDS: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning*, pages 5637–5664.

Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2015. Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web*, pages 625–635.

Severin Laicher, Sinan Kurtyigit, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. 2021. Explaining and improving BERT performance on lexical semantic change detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 192–202, Online. Association for Computational Linguistics.

Qi Liu, Matt J. Kusner, and Phil Blunsom. 2020. A survey on contextual embeddings. *arXiv*, abs/2003.07278.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. 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.

Kanishka Misra, Allyson Ettinger, and Julia Rayz. 2020. Exploring BERT's sensitivity to lexical cues using tests from semantic priming. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4625–4635, Online. Association for Computational Linguistics.

John Nerbonne and Wybo Wiersma. 2006. A measure of aggregate syntactic distance. In *Proceedings of the Workshop on Linguistic Distances*, pages 82–90, Sydney, Australia. Association for Computational Linguistics.

David Nigenda, Zohar Karnin, Bilal Zafar, Raghu Ramesha, Alan Tan, Michele Donini, and Krishnaram Kenthapadi. 2022. Amazon SageMaker Model Monitor: A system for real-time insights into deployed machine learning models. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Abhinav Ramesh Kashyap, Devamanyu Hazarika, Min-Yen Kan, and Roger Zimmermann. 2021. Domain divergences: A survey and empirical analysis. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1830–1849, Online. Association for Computational Linguistics.

Shruti Rijhwani and Daniel Preotiuc-Pietro. 2020. Temporally-informed analysis of named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7605–7617, Online. Association for Computational Linguistics.

Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. 2017. Data selection strategies for multi-domain sentiment analysis. *arXiv*, abs/1702.02426.

Abdus Saboor and Mohammad Abid Khan. 2010. Lexical-semantic divergence in Urdu-to-English example based machine translation. *6th International Conference on Emerging Technologies (ICET)*, pages 316–320.

Eyal Sagi, Stefan Kaufmann, and Brady Clark. 2009. Semantic density analysis: Comparing word meaning across time and phonetic space. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 104–111, Athens, Greece. Association for Computational Linguistics.

Ryoma Sato, Makoto Yamada, and Hisashi Kashima. 2022. Re-evaluating word mover's distance. In *Proceedings of the 39th International Conference on Machine Learning*.

Sidney J. Segalowitz and Korri C. Lane. 2000. Lexical access of function versus content words. *Brain and Language*, 75(3):376–389.

Xiaofei Sun, Yuxian Meng, Xiang Ao, Fei Wu, Tianwei Zhang, Jiwei Li, and Chun Fan. 2022. Sentence similarity based on contexts. *Transactions of the Association for Computational Linguistics*, 10:573–588.

Zoltán Gendler Szabó. 2022. Compositionality. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*, Fall 2022 edition. Metaphysics Research Lab, Stanford University.

Nina Tahmasebi, Lars Borin, and Adam Jatowt. 2021. Survey of computational approaches to lexical semantic change detection. In *Computational Approaches to Semantic Change*, pages 1–91. Language Science Press.

Liansheng Tang, Pang Du, and Chengqing Wu. 2010. Compare diagnostic tests using transformation-invariant smoothed ROC curves. *Journal of Statistical Planning and Inference*, 140(11):3540–3551.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488, Copenhagen, Denmark. Association for Computational Linguistics.

Geoffrey I. Webb, Roy Hyde, Hong Cao, Hai Long Nguyen, and Francois Petitjean. 2016. Characterizing concept drift. *Data Mining and Knowledge Discovery*, 30(4):964–994.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Ivan P. Yamshchikov, Viacheslav Shibaev, Nikolay Khlebnikov, and Alexey Tikhonov. 2021. Style-transfer and paraphrase: Looking for a sensible semantic similarity metric. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14213–14220.

Zihuiwen Ye, Pengfei Liu, Jinlan Fu, and Graham Neubig. 2021. Towards more fine-grained and reliable NLP performance prediction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3703–3714, Online. Association for Computational Linguistics.

A Appendix

Hyperparameter	**Value**
Learning rate decay	Linear
Warmup steps	10% of total
Learning rate	2e-5
Adam ϵ	1e-6
Adam β_1	0.9
Adam β_2	0.999
Attention dropout	0.1
Dropout	0.1
Weight decay	0.0
Batch size	32
Train steps	4 epochs

Table 2: Sentiment classification and NLI fine-tuning hyperparameters for the RoBERTa-base models in Section 4.2.

A.1 Model fine-tuning details

For each sentiment classification and NLI training domain in Section 4, we fine-tune a RoBERTa base-size model using the hyperparameters in Table 2 and the pre-trained RoBERTa model from Hugging Face, containing approximately 123M parameters (Liu et al., 2019; Wolf et al., 2020). Because there are only five training domains for MNLI, we run five fine-tuning runs per MNLI training domain; otherwise, we run one fine-tuning run per domain (43 domains for sentiment classification split by product category, 15 domains for sentiment classification split by review year). All models are fine-tuned using one Tesla V100 GPU, taking about two hours per model.

A.2 Efficient cosine distance computations

In Section 4.3, we compute the mean cosine distance between each evaluation example embedding $\boldsymbol{x}$ and all training example embeddings from D_{train}. Each example embedding is computed by taking the mean over all tokens in the example and the last two RoBERTa layers (before or after fine-tuning, as specified; Elango et al., 2022). Mean embedding cosine distances are also computed for individual tokens when quantifying lexical semantic change in Section 3.3. To avoid saving the embedding for each example in D_{train} and computing each cosine distance individually, we note that the mean pairwise cosine similarity between a set of vectors U and V is:

$$\underset{u\in U, v\in V}{\text{Mean}}\left(\cos(u,v)\right) = \frac{1}{|U||V|}\sum_{u\in U, v\in V}\frac{\langle u,v\rangle}{||u||\cdot||v||}$$

$$\begin{aligned}
&= \frac{1}{|U||V|}\sum_{u\in U}\sum_{v\in V}\left\langle \frac{u}{||u||}, \frac{v}{||v||}\right\rangle \\
&= \frac{1}{|U||V|}\sum_{u\in U}\left\langle \frac{u}{||u||}, \sum_{v\in V}\frac{v}{||v||}\right\rangle \\
&= \frac{1}{|U||V|}\left\langle \sum_{u\in U}\frac{u}{||u||}, \sum_{v\in V}\frac{v}{||v||}\right\rangle \\
&= \left\langle \frac{1}{|U|}\sum_{u\in U}\frac{u}{||u||}, \frac{1}{|V|}\sum_{v\in V}\frac{v}{||v||}\right\rangle \\
&= \left\langle \underset{u\in U}{\text{Mean}}\left(\frac{u}{||u||}\right), \underset{v\in V}{\text{Mean}}\left(\frac{v}{||v||}\right)\right\rangle
\end{aligned}$$

In other words, we only need to compute the dot product between the mean normed vector for U and V. For our uses, when computing the mean cosine distance between an example embedding $\boldsymbol{x}$ and all training example embeddings from D_{train}, we need only compute one minus the dot product between the normed $\boldsymbol{x}$ and the mean normed embedding over all examples in D_{train}. This way, we only need to store one vector (the mean normed embedding) for the entire training set, rather than one vector per training example.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations section (unnumbered).

☒ A2. Did you discuss any potential risks of your work?
Our work analyzes existing models (RoBERTa) on existing datasets, predicting their performance for out-of-domain data. Rather than introducing new models and risks, we hope that these results can be used to reduce the potential risks of applying existing models in cases where they might perform poorly.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and section 1 (Introduction).

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Datasets and models in Section 4 (Experiments): Amazon Reviews dataset, MNLI dataset, and Hugging Face RoBERTa model.

☑ B1. Did you cite the creators of artifacts you used?
Section 4 (Experiments).

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The datasets and models used are publicly available with citation.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 4 (Experiments).

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The datasets used are standard datasets, used as provided in their publicly available form.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4 (Experiments). More details can be found in the cited work corresponding to each dataset used.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4 (Experiments).

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

Section 4 (Experiments).

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4 (Experiments), Appendix A.1 (Model fine-tuning details).

□ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Not applicable. No hyperparameter search was used. RoBERTa hyperparameters are included in Appendix A.1 (Model fine-tuning details).

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4 (Experiments), Section 5 (Results), Appendix A.1 (Model fine-tuning details).

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4 (Experiments), Appendix A.1 (Model fine-tuning details). Used spaCy and the Hugging Face library.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.