Learning to Substitute Words with Model-based Score Ranking

Anonymous ACL submission

Abstract

Smart word substitution aims to enhance sentence quality by improving word choices, however current benchmarks rely on human-labeled data. Since word choices are inherently subjective, ground-truth word substitutions generated by a small group of annotators are often incomplete and likely not generalizable. To circumvent this issue, we instead employ a model-based score (BARTScore) to quantify sentence quality, thus forgoing the need for human annotations. Specifically, we use this score to define a distribution for each word substitution, allowing one to test whether a substitution is statistically superior relative to others. Further, we propose a loss function that directly optimizes the alignment between model predictions and sentence scores, while also enhancing the overall quality score of a substitution. Crucially, model learning no longer requires human labels, thus avoiding the cost of annotation while maintaining the quality of the text modified with substitutions. Experimental results show that the proposed approach outperforms both masked language models (BERT, BART) and large language models (GPT-4, LLaMA).

1 Introduction

In the current era of AI-driven technologies, the generation of natural language by machines has become a critical area of research within the field of natural language processing (NLP). The advent of advanced language models such as GPT (Ouyang et al., 2022) and LLaMa (Touvron et al., 2023a,b; Dubey et al., 2024) has brought about unprecedented improvements in generating coherent, contextually accurate text across a wide range of applications, including machine translation, summarization, and conversational agents. These models are revolutionizing industries by enabling machines to produce human-like text at scale, greatly expanding the possibilities for automation and interaction in both professional and creative domains.

As language generation capabilities improve, the need to refine and control the output of these models becomes more pressing. One key area in this direction is smart word suggestion (SWS) defined by Wang et al. (2023) as a task that focuses on enhancing writing by improving two aspects of NLP: identifying words (or tokens more generally) that if replaced can improve sentence quality, and suggesting alternatives for such words. Complementarily, Wang et al. (2023) introduced a dataset to benchmark models built for SWS. This task is essential in applications such as paraphrasing, machine translation, and writing assistance, where accurate word choice directly impacts the clarity and quality of the generated text. Despite the progress in full-text generation, word substitution presents unique challenges in maintaining grammatical correctness, contextual appropriateness, and semantic fidelity.

Previous SWS work can be divided into *task-specific* models and *prompt-based* methods. Task-specific models are trained directly to address the SWS task. One approach adapts models for lexical substitution (LS) (McCarthy and Navigli, 2007; Kremer et al., 2014), allowing them to generate substitutions for each word in a sentence. If the substitution differs from the original word, it is recognized and recorded as such. BERT-based (Zhou et al., 2019) and the LexSubCon (Michalopoulos et al., 2021) models were first explored in Wang et al. (2023). They also fine-tuned pretrained models like BERT (Devlin, 2018) and BART (Lewis, 2019), allowing them to handle SWS end-to-end. Importantly, these models were fine-tuned using supervised learning to provide suggestions for Wikipedia text using ground-truth substitutions obtained from a list of synonyms gathered from a thesaurus. Orthogonal to these machine learning techniques, rule-based methods that rely on paraphrasing databases, *e.g.*, PPDB (Pavlick et al., 2015)) or a thesaurus (*e.g.*, the Merriam-Webster thesaurus, have also been explored (Wang et al., 2023).

Alternatively, prompt-based techniques leveraging large language models (LLMs), such as GPT (Ouyang et al., 2022), have gained popularity in SWS. Provided that research in language generation has shifted from narrow tasks like substitution to generating coherent, contextually rich text, tasks such as SWS have become less of a priority. However, in general it is still key to enhance sentence quality, as precise word choices can significantly impact clarity, tone, and overall effectiveness in communication. Moreover, the success of these methods is highly dependent on prompt design.

From a different perspective, benchmarks for SWS predominantly rely on human-labeled data for both training and evaluating model performance. However, only a small group of annotators are recruited for data labeling, which can lead to a lack of diversity. For instance, in the SWS test dataset (Wang et al., 2023), 12,883 suggestions for target words are available of which 7,827 (60.8%) were created by a single annotator, only 1632 (12.67%) of suggestions were created by at least three annotators, and merely 633 (4.91%) of suggestions were created by more than three annotators. While scaling up the number of annotators could potentially mitigate this issue, it will significantly increases the costs of generating the annotations. In the context of substitution tasks, a large number of annotators is necessary to obtain a comprehensive representation of the substitution space, as evaluating them requires a diverse sample of high-quality alternatives. As a result, validation sets in such benchmarks may not fully capture model performance. Conversely, recent research has shown that model-based evaluation metrics align closely with human judgments (Lu et al., 2022; Yuan et al., 2021), offering a viable alternative to alleviate the drawbacks of human evaluations.

To address these challenges, we propose an approach that eliminates the dependency on human-labeled data. Our contributions are listed below:

- We identified a gap in the evaluation methods for substitution models, which rely on human annotators. We propose a model-based evaluation strategy based on the BARTScore to assesses SWS models without human annotations.
- We introduce an approach for generating and ranking token substitutions. Our method optimizes both the quality of the substitutions, via the model-based score, and the ranking of the substitutions relative to such score. To the best of our knowledge, this is the first time SWS has been tackled by jointly optimizing the association between predictions and quality scores.
- We redefined the substitution task by forgoing the need for supervised learning relying on human annotations. Thus, our method eliminates the need for costly human-labeled data while maintaining high performance in SWS tasks.
- Our experimental results demonstrate that the proposed approach outperforms both masked language models (BERT, BART) and LLMs (GPT-4, LLaMA) in the SWS task.

2 Related Work

Enhancing word usage is key feature for writing assistance. The SWS task in Wang et al. (2023) addresses it by emulating a real-world end-to-end writing scenario. Unlike traditional LS tasks (McCarthy and Navigli, 2007; Kremer et al., 2014), SWS requires systems to identify words that, if replaced, can improve sentence quality and suggest alternatives for such words. In the LS task, however, the word to be replaced is predefined and the model is only tasked with finding suitable substitutions without altering the overall meaning of the sentence. Further, in LS substitutions are typically lemmatized to match ground-truth labels, but in SWS they must be in the correct grammatical form.

Zhou et al. (2019) proposed a LS method using contextual word embeddings, modifying BERT with a dropout embedding policy to partially mask targets word and introduced a validation score based on the top four layers of BERT for candidate evaluation. Michalopoulos et al. (2021) integrated external knowledge from WordNet into BERT for LS, combining scores from BERT, WordNet gloss similarity, and sentence embeddings, along with the validation score from Zhou et al. (2019), to generate substitutes. ParaLS (Qiang et al., 2023) generated substitutes through a paraphrasing model, using a heuristics-based decoding strategy.

To adapt the substitution capabilities from LS to the SWS task, Wang et al. (2023) used an approach that allowed models to generate a substitutions for each word. If the substitution differs from the original word, it is recognized as the word the model wants to improve and recorded as a suggestion. Further, they fine-tuned pretrained models like BERT (Devlin, 2018) and BART (Lewis, 2019) to address the task in an end-to-end manner, and as an alternative to machine learning, rule-based methods using dictionaries (thesaurus) were also explored.

Recently, prompt-based methods have emerged as a powerful alternative for SWS (Wang et al., 2023), by leveraging the capabilities of LLMs such as GPT (Ouyang et al., 2022). These approaches shift the focus from traditional model fine-tuning to designing prompts that guide the model to produce desired outcomes. While this flexibility allows for easier adaptation to various tasks, the effectiveness of the method heavily relies on the prompt's structure. Despite the growing interest in generating fluent, contextually cohesive text with LLMs, prompt-based methods often overlook the importance of improving specific word choices. In contexts where sentence clarity and precision are key, refining word choices remains an important task that prompt-based approaches need to address.

Compared to traditional methods that rely heavily on human involvement, *e.g.*, via manual annotations, in our work, we utilize model-based sentence scoring, specifically, BARTScore to quantify the quality of sentences, thereby eliminating the human-associated costs. We align the model's output with the scores through a ranking loss, ensuring that substitutions with higher scores are more likely to be generated by the model, resulting in higher-quality substitutions. This approach forgoes the need for human-dependent supervised training, thus without the need for human intervention.

3 Methodology

In SWS, achieving a representative sample of high-quality alternatives requires a broad pool of annotators, as diverse substitution options are essential for accurate evaluation. However, relying solely on human annotators is thus not only costly but also inefficient. Below we introduce a method to train and evaluate token substitution models that does not require human annotations, by instead solely relying on model-based scoring functions.

Problem Definition Let X denote a sentence composed of N tokens, *i.e.*, $X = (x_1, \ldots, x_N)$. We aim to find a collection of K potential substitutes $\{\tilde{w}_k\}_{k=1}^{K}$ for token x_n, while estimating their likelihood given the sentence (*i.e.*, its context) as $p_\theta(x_n = \tilde{w}_k|X)$, where X is the original sentence and $p_\theta(\cdot)$ is a model for the conditional likelihood for token x_n parameterized by θ. Note that technically, if $p_\theta(\cdot)$ is a masked language model we should write $p_\theta(x_n = \tilde{w}_k|X_{\setminus x_n})$, where $X_{\setminus x_n}$ indicates that x_n has been masked out in X, and alternatively, if we use an (autoregressive) language model we should write $p_\theta(x_n = \tilde{w}_k|x_{<n}, Z)$, where $x_{<n}$ is the portion of the original sentence up to token $n-1$ and Z is a prompt. However, we use $p_\theta(x_n = \tilde{w}_k|X)$ in the following for notational simplicity. Based on these likelihood estimates, the token substitution rule is set as $x_n \leftarrow \tilde{w}_k$ if $p_\theta(x_n = \tilde{w}_k|X) > p_\theta(x_n = w_n|X)$, where w_n is the original value of x_n. Moreover, if multiple replacement candidates satisfy the substitution rule, we select $\tilde{w}_k$ according to $\text{argmax}_k\, p_\theta(x_n = \tilde{w}_k|X)$, and alternatively, if none satisfy it, we leave the token unchanged, *i.e.*, $x_n \leftarrow w_n$.

Importantly, $p_\theta(x_n = \tilde{w}_k|X)$ reflects the likelihood of token $\tilde{w}_k$ given its context X, rather than the quality of the sentence. So motivated, we also seek to align such estimates with a score $M(x_n = \tilde{w}_k|X)$ quantifying the quality of sentence X with x_n substituted with $\tilde{w}_k$. For simplicity, in the following we denote the sentence X with x_n replaced with the k-th candidate, $\tilde{w}_k$, as $\tilde{X}_k = (\tilde{x}_1, \ldots, \tilde{x}_N)$. Further, to make our objective scalable and general, the score $M(x_n = \tilde{w}_k|X)$ ought not be obtained via human feedback, but via a *black-box model* with which we can score sentences, but through which we cannot learn, *i.e.*, gradients cannot be propagated through it to obtain learning signals.

Below we start by showing how we can use a model-based score to statistically characterize the suitability of a candidate as an alternative to the substitution rule introduced above, and then present an approach to estimate $P(x_n = \tilde{w}_k|X)$ while accounting for scores $M(x_n = \tilde{w}_k|X)$.

3.1 Statistic for Model-based Scores

Model-based Score Using automated scoring methods to quantify text generation quality without relying on human annotators can reduce costs associated with human labeling. Model-based scoring approaches such as BLEURT (Sellam et al., 2020), BERTScore (Zhang et al., 2019), GPTScore (Fu et al., 2023), and BARTScore (Yuan et al., 2021), have proven to be effective while closely aligning with human evaluation results. BARTScore is one of the most popular metrics for the evaluation of text generation. It has been shown that BARTScore outperforms other metrics such as BERTScore and BLEURT (Yuan et al., 2021), while also being more efficient than GPTScore. Further, it has been used also as a ranking tool for substitution tasks, achieving good results (Qiang et al., 2023).

The BARTScore is a metric for universal natural language generation evaluation (Yuan et al., 2021).

It leverages the conditional likelihood from a pre-trained BART model (Lewis, 2019) to assess the quality of generated text. Since BART is an autoregressive model, the log-likelihood of tokens in a sentence is obtained one at the time conditioned on tokens before it. Specifically, we write

$$M(\tilde{X}_k) = \sum_{n=1}^{N} \log p_{\hat{\theta}}\left(\tilde{x}_n \mid \tilde{x}_{<n}, X\right) , \quad (1)$$

where $\hat{\theta}$ represents the parameters of a pre-trained model (in this case, BART), $\tilde{x}_n$ is the n-th token of the modified sentence $\tilde{X}_k$ which we wish to score, $\tilde{x}_{<n}$ represents the first $n-1$ tokens of $\tilde{X}_k$, and X is the original (unmodified) sentence. In (1), the BARTScore $M(x_n = \tilde{w}_k|X)$ has been simplified for notational convenience to $M(\tilde{X}_k)$, *i.e.*, the score of modified sentence $\tilde{X}_k$ in which a token has been replaced with the k-th candidate from potential substitutes $\{w_k\}_{k=1}^{K}$. For now, we assume that such candidates are readily available, however, in the next section we will propose a learning strategy to train a model to generate substitutes given X.

Model-based Score Statistic One potential issue with model-based scores including the BARTScore in (1) is that there is no clear standard to determine what score (or score range) qualifies as "good". It is clear that though scores can be used for pairwise or aggregate comparisons, the magnitude of the difference between two scores is not meaningful without a reference score distribution.

This underscores that numerical scores alone are insufficient to fully capture the significance of a score difference. Inspired by statistical hypothesis testing (Moore, 1999), we propose constructing a reference (null) distribution of $M(\tilde{X}_k)$ by sampling a collection of substitutes for a given token and then calculating their model-based score. By determining where $M(\tilde{X}_k)$ lies within this empirical distribution, we can more reliable assess the quality of a substitute. Since sampling candidates uniformly at random from a natural language vocabulary or from empirical token frequencies will be hugely inefficient, instead we use a model to generate substitute candidates. In this manner, we can evaluate the relative quality of the generated substitutes compared to other candidates from the model's own distribution. This constitutes a principled framework for assessing the quality of a model in generating token substitutes. Next, we introduce a hypothesis test to estimate a p-value for the significance of a candidate substitute relative to a reference (empirical) substitute distribution.

Given a model we wish to evaluate, let $\{w_k\}_{k=1}^{K_s}$ be a set of K_s candidates for sentence X at a given position produced by such model, *e.g.*, a masked language model like BERT (Devlin, 2018). Note that in general, K_s is a number much larger than the size of potential substitutes K described before. We can evaluate the quality of, for instance, the top candidate w_1 (assuming they are ordered) as the proportion of times it produces a score that is larger than using all other candidates, $\{w_k\}_{k=2}^{K_s}$. This frequency or p-value is formally written as

$$p_{\tilde{X}_1} = \frac{1}{K} \sum_{k=2}^{K_s} I[M(\tilde{X}_k) > M(\tilde{X}_1)] , \quad (2)$$

where $I[\cdot]$ is the indicator function and the $\{w_k\}_{k=1}^{K_s}$ producing modified sentences $\{\tilde{X}_k\}_{k=1}^{K_s}$ are obtained from a model $p_\theta(x_n = \tilde{w}_k|X)$, which defines the reference distribution. In standard hypothesis testing fashion, we reject the hypothesis that candidate w_1 has the same expected score as the reference distribution if $p_{\tilde{X}_1} < \alpha$, for a significance level α, which we set to $\alpha = 0.01$ in our experiments. Since calculating model-based scores such as BARTScore is computationally expensive, we trade-off statistical accuracy with computational efficiency and let $K_s = 1000$ in the experiments.

3.2 Preference-Aware Learning

Now that we have constructed a statistic based on model-based scores, in principle, we seek to train a model such that for each token x_n, the resulting $p_{\tilde{X}_k}$ from (2) for a given substitute candidate w_k producing $\tilde{X}_k$ is as small as possible. According to (2), it is desirable for $M(\tilde{X}_1)$ to be larger than $M(\tilde{X}_{\hat{k}})$ for $\{w_k\}_{k=1}^{K}$ candidates sampled from model $p_\theta(x_n|X)$. This means that effectively we require the model being such that its outputs align, in likelihood, with the model-based score, BARTScore here. Consequently, it is desirable to learn parameters θ so $p_\theta(x_n = w_k|X)$ is ranked consistent to $M(\tilde{X}_k)$, which in turn will make $p_{\tilde{X}_k} \to 0$ in (2) for $k \to 1$. Below we consider two approaches to accomplish this: margin ranking (Liu et al., 2023b; Chern et al., 2023; Liu et al., 2022) and direct preference optimization (DPO) loss (Rafailov et al., 2024).

Ranking Optimization Ranking has been used as optimization objective in many tasks including text summarization (Chern et al., 2023; Liu et al., 2022). Maximum likelihood estimation based on the standard cross-entropy loss can be effective at

satisfying (2) by setting it as a classification problem by defining a vector of labels y where $y_k = 1$ if $M(\tilde{X}_k)$ is maximum among $\{M(\tilde{X}_k)\}_{k=1}^K$ or $y_k = 0$ otherwise. However, it does not take into account the ordering of the model-based scores or their magnitude differences. So motivated, we consider the following margin ranking (MR) loss (Liu et al., 2023b; Chern et al., 2023; Liu et al., 2022)

$$\mathcal{L}_{\text{MR}} = \sum_k^K \sum_{j>k}^K \max(0, s_j - s_k + \lambda_{jk}) \,, \quad (3)$$

where $\{s_k\}_{k=1}^K$ are the logits from model $p_\theta(x_n = w_k|X)$, *i.e.*, before applying the softmax function, and we have sorted them such that $s_k > s_j$ if $M(\tilde{X}_k) > M(\tilde{X}_j)$ for $i, j = 1, \ldots, K$. Further, we set the *margin* $\lambda_{jk} = \lambda \times (j - k)$ for some hyperparameter λ, which in the experiments is set via cross-validation. Intuitively, (3) encourages the model to make predictions whose outputs $\{p_\theta(x_n = w_k|X_{\setminus x_n})\}_{k=1}^K$ are consistent in order with $\{M(\tilde{X}_k)\}_{k=1}^K$, by penalizing pairwise order mismatches, while also enforcing predictions to be distanced by a fixed margin to improve robustness. The latter is justified by extensive results from the margin learning literature (Smola, 2000).

Improving Model-based Scores One unintended consequence of the loss function in (3) is that though it encourages model predictions to be aligned with model-based scores, it does so regardless of their values. This is so because only the order of $\{M(\tilde{X}_k)\}_{k=1}^K$ is considered in (3). In practice, we have observed that (3) effectively improves the ranking of predictions from the model, but it does so at the expense of producing predictions that have, in average, lower model-based scores relative to a reference model, *e.g.*, the pre-trained model used as initialization for the refinement of $\{p_\theta(x_n = w_k|X)\}_{k=1}^K$. To address this issue, we consider two approaches, one that seeks to improve the weighted average of model-based scores and another that maximizes the model-based score of the top prediction from the model relative to that of the reference model. Specifically, we write

$$\mathcal{L}_{\text{AS}} = -\sum_k^K h(s_k) M(\tilde{X}_k) \,, \quad (4)$$

$$\mathcal{L}_{\text{BS}} = \max(0, (M(X) - M(\tilde{X}_1)) f(s_1)) \,, \quad (5)$$

where $h(\cdot)$ is the softmax function, s_k is the logit corresponding to $p_\theta(x_n = w_k|X)$, and $M(X)$ and $M(\tilde{X}_k)$ are the model-based scores of the original and modified sentence, respectively. Recall that for $M(\tilde{X}_1)$, the token of interest x_n in X has been modified to the top prediction w_1 from the model. Conceptually, $\mathcal{L}_{\text{AS}}$ in (4) seeks to maximize the (weighted) average of model-based scores from K predictions, while $\mathcal{L}_{\text{BS}}$ aims to improve the model-based score of the top prediction wrt that of a reference model.

We then combine the margin ranking loss in (3) with the score-improving losses in (4) or (5) as

$$\mathcal{L}_{\text{MR+AS}} = \mathcal{L}_{\text{MR}} + \gamma \mathcal{L}_{\text{AS}} \,, \quad (6)$$

$$\mathcal{L}_{\text{MR+BS}} = \mathcal{L}_{\text{MR}} + \gamma \mathcal{L}_{\text{BS}} \,, \quad (7)$$

where γ is a hyperparameter trading off ranking or model-based score improvement. In the experiments we will compare these two approaches to determine empirically whether is better to attempt to improve the scores of K predictions from the model as opposed to improving the score of one of the predictions from the model (the most likely) relative to a baseline prediction obtained from a reference model. Note that is tempting to use (5) with all predictions, not just the first one, however, we found empirically that it considerably increases the computational cost without significant performance gains. The cost overhead is caused mainly by the need to evaluate the model-based score $2K$ times rather than just 2 in (5) or K in (4).

Direct Preference Optimization Direct Preference Optimization (DPO) (Rafailov et al., 2024) is an efficient technique for aligning large language models with human feedback, which gained popularity due to its simplicity. For instance, it has demonstrated to be effective in chat benchmarks (Tunstall et al., 2023; Zheng et al., 2023). In our case, the model-based score serves as proxy for human feedback DPO (under the Plackett-Luce model) and is formally expressed as

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}\left[\log \prod_{k=1}^K \frac{\exp(\delta r_k)}{\sum_{j=k}^K \exp(\delta r_j)}\right] \,, \quad (8)$$

where $r_k = \log p_\theta(\tilde{X}_k) - \log p_{\hat{\theta}}(\tilde{X}_k)$, the expectation is over $\{\tilde{X}_1, \ldots, \tilde{X}_K, X\}$, and θ and $\hat{\theta}$ denote the parameters of the model being trained and that used for reference, respectively. Accordingly, only θ are updated while learning while $\hat{\theta}$ are kept fixed.

Since the magnitude of the logits drops significantly as $k \to K$, we found that the sum in the denominator of (8) weakens the loss. Therefore, we removed the sum and instead compared the k-th and the $(k+1)$-th substitution, rather than comparing it with all K values. Further, we approximate

$\log p_\theta(\tilde{X}_k)$ with its logit s_k, and let $\delta = 1$ for convenience. Then (8) simplifies (for a single token in sentence X) to

$$\mathcal{L}_{\text{DPO}^*} = -\sum_{k=1}^{K-1} \left(s_k - \hat{s}_k - s_{k+1} + \hat{s}_{k+1}\right) , \quad (9)$$

where s_k and $\hat{s}_k$ denote the logit of the k-th substitution from the model being trained and the reference model, respectively.

We also extend DPO (under the Bradley-Terry model) (Rafailov et al., 2024) to multiple substitute candidates, where each candidate is compared with the next in the ordered list of substitute candidates. We write (for a single token in sentence X)

$$\mathcal{L}_{\sigma\text{DPO}^*} = \quad (10)$$
$$-\sum_{k=1}^{K-1} \log \sigma \left(s_k - \hat{s}_k - s_{k+1} + \hat{s}_{k+1}\right) ,$$

from which we see that the only difference between (10) and (9) is that the comparison of logit values in the former is scaled with the log-logistic function. See Appendix A for a derivation of both losses.

4 Experiments

Below we illustrate the problem with evaluating with human annotations, then we present an ablation study comparing the optimization approaches in Section 3.2 and a benchmark comparing the proposed model to MLM and LLM approaches in terms of model-based score alignment, average score and the statistic in (2). Further, we explore the performance of top-2 predictions, *i.e.*, when the top prediction is the original token, and results comparing LLMs with and without prompts encouraging token substitutes to be ranked by quality.

Datasets We consider four datasets. The SWS dataset by Wang et al. (2023) consists of three sets, validation and test labeled by human annotators, and an artificially generated training set. The training dataset was constructed from Wikipedia sentences, with labels obtained using a combination of PPDB (Pavlick et al., 2015) and Merriam-Webster thesaurus. We also consider two traditional lexical substitution datasets, LS07 (McCarthy and Navigli, 2007) and LS14 (Kremer et al., 2014), which use lemmatized human annotations as ground truth substitutions. Further, we also consider a general dataset without ground-truth substitutions. Specifically, we use XSum (Narayan et al., 2018), a summarization dataset consisting of BBC news articles and corresponding summaries. In the Appendix we show Table 6 summarizing these four datasets.

Baselines We compare our method with three classes of baseline models. *Masked language models (MLMs):* BERT-base-uncased with original MLM head (BERT-naive) (Devlin, 2018). BERT-spsv (Zhou et al., 2019), as a representative LS model. BERT-SWS and BART-SWS, both of which were fine-tuned on the SWS training dataset (Wang et al., 2023). *Rule-based models:* The rule-based approach introduced by Wang et al. (2023), which leverages a thesaurus. *Prompt-based large language models (LLMs):* We leverage the power of pretrained LLMs via prompts to generate token substitutions. We consider two popular choices, GPT (Ouyang et al., 2022) and LLaMa (Touvron et al., 2023a,b; Dubey et al., 2024). Specifically, GPT-4o and Llama-3.1-8B-Instruct. Prompt-based techniques have gained popularity due to their flexibility and effectiveness in generating high-quality, contextually appropriate substitutions without requiring task-specific tuning (Liu et al., 2023a). We designed both ranking and non-ranking prompts, which we show in Appendix B.

Evaluation Metrics We use the cosine similarity (CS) to measure the correlation between model predictions and model-based scores, *i.e.*, BARTScores. Since BARTScore produces log-likelihoods, we log-transform model predictions accordingly. We also consider the quality of the substitutes $\{\tilde{w}_k\}_{k=1}^K$. First we get the score ratio between the sentence with and without substitution and then calculate the average of BARTScore ratios as $\text{ABR} = 1/K \sum_{k=1}^{K} M(\tilde{X}_k)/M(X)$.

Implementation For our model, we fine-tuned the BERT-base-uncased model (Wang et al., 2023) on 100k randomly sampled sentences from the SWS training dataset using one Nvidia A100 GPU. For each sentence, we randomly selected 5 tokens, and for each token, we chose $K = 5$ candidates from the model's output to form a candidate pool and compute the training loss. The model was fine-tuned over 5 epochs. For DPO, we duplicated the same BERT model and weights at initialization, freezing one as the reference while only updating the weights of the other during training. During testing, we used the same token selection strategy, randomly selecting 5 tokens per sentence and choosing the top 5 candidates for each. Hyperparameter settings are described in Appendix C.

Illustrative Example We stratify tokens in the SWS test set into five groups according to whether

	Not change	Agreement	Disagreement
Change	4125 (0.74)	631 (0.11)	830 (0.15)
Not change	14138 (0.94)	969 (0.06)	

Table 1: Token changes in SWS test data. Rows are for annotator changes and columns indicate model changes and if they are in agreement with the annotator. In parenthesis are proportions relative to the row totals.

	SWS		LS07		LS14		XSum		AVG	
	CS	ABR	CS	ABR	CS	ABR	CS	ABR	CS	ABR
MR	**0.99**	0.82	0.93	0.81	0.93	0.78	0.93	0.79	0.94±0.024	0.8±**0.012**
DPO	0.91	0.87	0.9	0.86	0.91	0.83	0.85	0.82	0.89±0.025	0.85±0.025
DPO*	0.92	0.87	0.91	0.86	0.91	0.82	0.87	0.83	0.9±0.019	0.85±0.021
σDPO*	0.93	0.89	0.93	0.86	0.92	0.84	0.88	0.84	0.92±0.021	0.86±0.02
MR+BS	0.94	**0.90**	0.95	**0.87**	0.95	**0.86**	0.95	**0.86**	0.94±0.005	**0.87**±0.015
MR+AS	0.98	0.890	**0.99**	0.85	**0.99**	0.83	**0.99**	0.84	**0.99±0.002**	0.86±0.021

Table 2: Ablation CS and ABR metrics. Figures for each dataset are medians over all token predictions. The last column shows averages over all datasets with standard deviations. The best results are highlighted in **bold**.

	SWS		LS07		LS14		XSum		AVG	
	CS	ABR	CS	ABR	CS	ABR	CS	ABR	CS	ABR
BERT-naive	0.92	0.87	0.92	0.83	0.92	0.82	0.93	0.81	0.92±0.003	0.83±0.018
BERT-spsv	0.93	0.82	0.93	0.80	0.93	0.74	0.93	0.74	0.93±0.002	0.78±0.038
BERT-SWS	0.92	0.84	0.92	0.81	0.93	0.78	0.93	0.79	0.93±0.003	0.81±0.02
BART-SWS	0.93	0.84	0.92	0.83	0.92	0.77	0.93	0.80	0.93±0.003	0.81±0.028
Rule-based	0.93	**0.90**	0.92	0.83	0.93	0.82	0.93	0.83	0.93±0.001	0.85±0.024
GPT-4o	0.95	0.89	0.95	**0.86**	0.96	0.82	0.96	**0.85**	0.95±0.002	**0.86±0.025**
LLaMa	0.93	0.86	0.94	0.84	0.93	0.76	0.94	0.79	0.94±0.003	0.81±0.040
MR+AS	**0.98**	**0.90**	**0.99**	0.85	**0.99**	**0.83**	**0.99**	0.84	**0.99±0.002**	**0.86±0.021**

Table 3: Benchmark median CS and ABR metrics. The last column shows averages over all datasets with standard deviations. The best results are highlighted in **bold**.

they were substituted by human annotators and/or the model. These groups are: change agreement (CA: both model and the human annotator agreed on the change), change disagreement (CD: both model and the annotator suggested a change, but the model substitution did not match that of the annotator), no change agreement (NCA: the model and the annotator kept the token unchanged), only model changed (OMC: the model suggested a change, but the annotator did not find it necessary), and only annotator changed (OAC: the annotator suggested a change, but the model did not).

Results in Table 1 show that most tokens remain unchanged (NCA: 94%) by both model (BERT-SWS) and annotator. The proportion of annotator changes for which the model agrees or disagrees is similar, CA: 11% *vs.* CD: 15%, respectively. Interestingly, most annotator changes are not captured by the model (OAC: 74%), which is consistent with results in Wang et al. (2023) for a variety of models, and OMC: 6% tokens not changed by the annotator are changed by the model. These results underscore that i) a model is unlikely to match human annotations when these are largely subjective and incomplete; and ii) cases in which changes made by the model do not match the human annotator cannot be considered errors. These are consequences of having a few annotators each selecting which tokens need to be changed.

This is further illustrated below in Table 4 where we show that though tokens for which both model and annotator agree have better model-based statistics, there is also a non-insignificant portion for which model and annotator disagree, but the model-based statistic suggests that the model changes are of good quality. Some of these examples are shown in Table 7 of the Appendix.

Ablation Study Section 3.2 introduced two approaches to align the predictions of a trained model with model-based scores, namely MR in (3) and DPO in (8), (9) and (10). For the former, we consider two ways of improving the model-based scores, via a weighted average in (6) and improvement relative to a reference model in (7). Results in Table 2 show results comparing i) the alignment of model predictions with model-based scores using median CS, and ii) the average model-based scores using median ABR. Distributions of CS and ABR for all tokens are shown in the Appendix Figures 1 and 2. We see that i) the MR loss alone is not sufficient to deliver the best CS and produces the worst ABR; ii) DPO variants improve the ABR relative to MR, but not in terms of CS; c) MR+BS and MR+AS produce the best overall ABR and CS, respectively, however, MR+AS seems to provide the best trade-off between the two performance metrics, while outperforming both MR and DPO. Consequently, in the following experiments we will focus on MR+AS.

Model Benchmark Next, we evaluate MLMs (BERT, BART) and LLMs (GPT-4o, LLaMa) on all datasets using median CS and ABR as metrics (distributions are shown in the Appendix Figures 1 and 2). Table 3 shows that MR+AS outperforms both MLMs (BERT, BART) and LLMs (GPT-4o, LLaMA), with top performances in terms of CS in all datasets and top ABR in two of the datasets (SWS and LS14). Also, when accounting for results variation across datasets, we see that MR+AS significantly outperforms the others in terms of CS, while being comparable with GPT-4o in terms of ABR. Importantly, for the latter, MR+AS and GPT-4o perform markedly better than other approaches.

Model-based Statistic Benchmark After demonstrating the effectiveness of our model in terms of CS and ABR, we now use the model-based

	CA	CD	OMC
BERT-naive	0.646	0.389	0.612
BERT-spsv	0.709	0.442	0.491
BERT-SWS	0.204	0.118	0.078
BART-SWS	0.225	0.139	0.121
Rule Based	0.602	0.241	0.344
GPT-4o	0.761	0.563	0.661
LLaMa	0.700	0.517	0.603
MR+AS	**0.961**	**0.917**	**0.810**

Table 4: Proportion of p-values below the significance threshold ($\alpha = 0.01$) on the SWS test data.

	SWS	LS07	LS14	XSum	AVG
BERT-naive	0.88	0.84	0.82	0.81	0.84± 0.025
BERT-spsv	0.82	0.80	0.73	0.73	0.77± 0.040
BERT-SWS	0.85	0.82	0.78	0.79	0.81± 0.028
BART-SWS	0.82	0.81	0.74	0.77	0.79± 0.035
Rule Based	0.89	0.84	0.82	0.85	0.85± 0.025
GPT-4o	0.88	0.86	0.80	0.85	0.85± 0.028
LLaMa	0.86	0.83	0.77	0.79	0.81± 0.035
MR+AS	**0.91**	**0.87**	**0.85**	**0.85**	**0.87 ± 0.024**

Table 5: Median BARTScore ratios for top-2 candidates. The last column shows averages over all datasets with SDs. The best results are highlighted in **bold**.

statistic introduced in Section 3.1. Table 4 shows the proportion of model predictions that meet the significance threshold ($\alpha = 0.01$) for three groups of tokens: CA, CD and OMC. We do not present results for the other two groups (NCA and OAC) because when using rule- and LLM-based models, we get no candidates for non-substitutions. Further, for these models we may not always get the same number of substitute candidates per token, thus we use their median (3 candidates) across all tokens in the SWS test set as the number of candidates for all other models when calculating the statistic, *i.e.*, $K_s = 3$ in (2). In the Appendix Table 8 we show an extended table showing the NCA and OAC groups and $K_s = 1000$ for MLM-based models (ours included) and Figures 3-6 with p-value distributions for all models.

Ideally, we want proportions in Table 4 to be close to one, indicating that the top candidate is of better quality than the alternatives. Effectively, results demonstrates that our model (MR+AS) not only produces the largest proportions, but it does so irrespective of the group (CA, CD or OMC). This suggests that substitutions made by our model are of good quality (using BARTScore as proxy for quality) even if they do not agree with the human annotator. Moreover, in general, LLMs are better than the other MLM- and rule-based alternatives.

It is worth noting that LLM-based models make considerably more substitutions than our model. In the Appendix Table 9 we show that GPT-4o makes 24.4%, 24.8% and 11.5% changes in groups CA, CD and OMC, respectively, compared to 5.5%, 5.2% and 2.7% by our MR+AS. This suggests a trade-off between quality and quantity of substitutions between MR+AS and GPT-4o, respectively.

Top-2 Candidate Performance Provided that the substitution rates by our model are low relative to others as discussed above, we now examine the quality of the top-2 substitute. Note that in MLM-based models a change is produced only when the top substitute candidate is different than the original token and that for LLM-based models the top-2 candidate is only available is the model considers a substitution has to be made and more than one candidates are provided. To quantify the quality of the top-2 candidates, we randomly selected five tokens from each sentence where the top-2 candidate is available and calculate the BARTScore ratio between the sentence modified with the top-candidate and that of the original (unchanged) sentence. We then averaged these ratios provide a summary of the quality of top-2 candidates. Results in Table 5 indicate that MR+AS achieved the best top-2 candidate quality across all four datasets followed (in average) by GPT-4o and the rule-based model.

Encouraging ranking in LLMs It is well known that prompting LLMs to provide ordered responses is challenging. We tried prompts with and without a specific request for ordered substitute candidates (see Appendix B for details). We found that both GPT and LLaMa produced slightly better median CS values without the prompt specifying ranked candidates (0.954 and 0.943, respectively) relative to the alternative (0.953 and 0.940, respectively). Importantly, these are significantly lower than the median CS value for MR+AS (0.983). CS distributions for all models can be found in the Appendix Figure 7. We note that it may be possible to optimize the prompt for better ranked results, however, this is beyond the scope of our work.

5 Discussion

This paper addressed the problem of training and evaluating SWS models without requiring human annotations. Specifically, we leveraged a model-based score (BARTScore) to define evaluation metrics and to serve as a label-proxy for our loss function. Extensive experimental results demonstrate the effectiveness of our approach relative to both MLM- and LLM-based models.

Limitations

The proposed method leverages BARTScore as a sentence scoring mechanism, essentially acting as a proxy for human annotators in guiding model optimization during preference-aware learning. However, we cannot eliminate the possibility that BARTScore may not always accurately quantify the quality of token substitutions. While BARTScore has shown effectiveness in our experiments, it could be readily replaced by alternative scoring metrics deemed more reliable or better suited to other word substitution contexts. We leave this exploration as interesting future work.

Ethical Considerations

In the process of developing and deploying smart word suggestion systems it is crucial to consider the broader ethical implications of relying on automated tools for language optimization. These systems may inadvertently reinforce biases present in training data, leading to unintended or inappropriate word suggestions. Moreover, the replacement of human judgment with automated scoring models like BARTScore can risk overlooking or misrepresenting nuances in language, particularly in sensitive contexts. Therefore, we recommend ongoing human oversight and continuous evaluation of the model's output to ensure fairness, inclusivity, and alignment with ethical standards.

References

I-Chun Chern, Zhiruo Wang, Sanjan Das, Bhavuk Sharma, Pengfei Liu, Graham Neubig, et al. 2023. Improving factuality of abstractive summarization via contrastive reward learning. *arXiv preprint arXiv:2307.04507*.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Gerhard Kremer, Katrin Erk, Sebastian Padó, and Stefan Thater. 2014. What substitutes tell us-analysis of an "all-words" lexical substitution corpus. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 540–549.

M Lewis. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. Brio: Bringing order to abstractive summarization. *arXiv preprint arXiv:2203.16804*.

Yixin Liu, Kejian Shi, Katherine S He, Longtian Ye, Alexander R Fabbri, Pengfei Liu, Dragomir Radev, and Arman Cohan. 2023b. On learning to summarize with large language models as references. *arXiv preprint arXiv:2305.14239*.

Qingyu Lu, Liang Ding, Liping Xie, Kanjian Zhang, Derek F Wong, and Dacheng Tao. 2022. Toward human-like evaluation for natural language generation with error analysis. *arXiv preprint arXiv:2212.10179*.

Diana McCarthy and Roberto Navigli. 2007. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*, pages 48–53.

George Michalopoulos, Ian McKillop, Alexander Wong, and Helen Chen. 2021. Lexsubcon: Integrating knowledge from lexical resources into contextual embeddings for lexical substitution. *arXiv preprint arXiv:2107.05132*.

Jason H Moore. 1999. Bootstrapping, permutation testing and the method of surrogate data. *Physics in Medicine & Biology*, 44(6):L11.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational*

Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 425–430.

Jipeng Qiang, Kang Liu, Yun Li, Yunhao Yuan, and Yi Zhu. 2023. Parals: lexical substitution via pretrained paraphraser. *arXiv preprint arXiv:2305.08146*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Alexander J Smola. 2000. *Advances in large margin classifiers*. MIT press.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Chenshuo Wang, Shaoguang Mao, Tao Ge, Wenshan Wu, Xun Wang, Yan Xia, Jonathan Tien, and Dongyan Zhao. 2023. Smart word suggestions for writing assistance. *arXiv preprint arXiv:2305.09975*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. Bert-based lexical substitution. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 3368–3373.

A DPO Derivation Details

DPO* Direct Preference Optimization (DPO) (Rafailov et al., 2024) is an efficient technique for aligning large language models (LLMs) with human feedback, which gained popularity due to its simplicity. For instance, it has demonstrated to be effective in chat benchmarks (Tunstall et al., 2023; Zheng et al., 2023). In our case, the model-based score serving as proxy for human feedback DPO (under the Plackett-Luce Model) is written as

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}\left[\log \prod_{k=1}^{K} \frac{\exp(\delta r_k)}{\sum_{j=k}^{K} \exp(\delta r_j)}\right] , \quad (11)$$

where $r_k = \log p_\theta(\tilde{X}_k) - \log p_{\hat{\theta}}(\tilde{X}_k)$, the expectation is over $\{\tilde{X}_1, \ldots, \tilde{X}_K, X\}$, and θ and $\hat{\theta}$ denote the parameters of the model being trained and that used for reference, respectively. Accordingly, only θ are updated while learning while $\hat{\theta}$ are kept fixed.

Since the magnitude of the logits drops significantly as $k \to K$, we found that the sum in the denominator of (11) weakens the loss. Therefore, we removed the sum and instead compared the k-th and the $(k+1)$-th substitution, rather than comparing it with all K values. The Derivation of DPO* is shown below

$$\begin{aligned}
\mathcal{L}_{\text{DPO}*} &= -\log \prod_{k=1}^{K-1} \frac{\exp(\delta r_k)}{\exp(\delta r_{k+1})} \quad (12)\\
&= -\sum_{k=1}^{K-1} \log \frac{\exp(\delta r_k)}{\exp(\delta r_{k+1})}\\
&= -\sum_{k=1}^{K-1} (\delta r_k - \delta r_{k+1})\\
&= -\sum_{k=1}^{K-1} \delta \left(\log \frac{p_\theta(\tilde{X}_k)}{p_{\hat{\theta}}(\tilde{X}_k)} - \log \frac{p_\theta(\tilde{X}_{k+1})}{p_{\hat{\theta}}(\tilde{X}_{k+1})}\right) .
\end{aligned}$$

Further, we approximate $\log p_\theta(\tilde{X}_k)$ with its logit s_k, and let $\delta = 1$ for simplicity. Then (12) simplifies (for a single token in sentence X) to

$$\mathcal{L}_{\text{DPO}*} = -\sum_{k=1}^{K-1} (s_k - \hat{s}_k - s_{k+1} + \hat{s}_{k+1}) ,$$

where s_k and $\hat{s}_k$ denote the logit of the k-th substitution from the model being trained and the reference model, respectively.

σDPO* We also extend DPO (under the Bradley-Terry model) (Rafailov et al., 2024) to multiple substitute candidates, where each candidate is compared with the next in the ordered list of candidates. We write for a single token in sentence X

$$\mathcal{L}_{\sigma\text{DPO}*} = -\sum_{k=1}^{K-1} \log \sigma (\delta r_k - \delta r_{k+1}) . \quad (13)$$

We approximate $\log p_\theta(\tilde{X}_k)$ with its logit s_k, and let $\delta = 1$ for simplicity. Then in (13) simplifies (for a single token in sentence X) to

$$\begin{aligned}
&\mathcal{L}_{\sigma\text{DPO}*} =\\
&\quad -\sum_{k=1}^{K-1} \log \sigma (s_k - \hat{s}_k - s_{k+1} + \hat{s}_{k+1}) ,
\end{aligned}$$

from which we see that the only difference between (10) and (9) (in the main paper) is that the comparison of logit values in the former is scaled with the log-logistic function. See Appendix for a derivation of both losses.

B LLM Prompts

Prompt <u>without</u> order:

In the following sentence, please give some suggestions to improve word usage. Please give the results with the JSON format of "original word": ["suggestion 1", "suggestion 2"]. The 'original word' should include all words that can be improved in the sentence, directly extracted from the sentence itself. [s]

Prompt <u>with</u> order:

In the following sentence, please give some suggestions to improve word usage. Please give the results with the JSON format of "original word": ["suggestion 1", "suggestion 2"]. The 'original word' should include all words that can be improved in the sentence, directly extracted from the sentence itself, and ***the suggestions should be ranked in order of the degree of improvement, from the most effective to the least****. [s]*

where [s] is the sentence.

At times, the model may fail to provide the correct JSON format, making it difficult to extract the intended answer. In such cases, it is often necessary to query the model multiple times to obtain a valid JSON output.

C Model Hyperparameters

Both of BERT_SWS reproduction and model optimization are: epochs $= 5$, batch size $= 64$, learning rate $= 0.0007$, max norm $= 1e-5$ for clipped gradient norm, dropout rate $= 0.1$, $\lambda = 0.5$, and $\gamma = 1$.

Data	Task	Training	Validation	Test
SWS	SWS	3,909,650	200	800
LS07	Lexical Substitution	-	299	1710
LS14	Lexical Substitution	-	892	1569
XSum	Summarization	204045	11332	11,334

Table 6: Datasets summary.

CD	While being saddened by my friend's suffering, I also gradually became aware of the critical role of law in our daily life. Annotations: Human={"integral", "important"}, Model={"crucial"}
	The residential theory argues that there are houses in the Chaco structures, and the structure is big enough for hundreds of people. Annotations: Human={"discussed", "states", "reasons", "discusses"}, Model={"claims"}
	As I planned well for the whole day, I would not make mistakes such as doing things relating to a single subject in a complete morning or afternoon, which would make my head dizzy and I could not focus all my attention on my work. Annotations: Human={"single", "whole"}, Model= {"full"}
	By contrast, the professor refutes this idea and insists that it will not be that profitable when the cost is taken into consideration. Annotations: Human={"urges", "claims"} Model= {"argues"}
	For example, in traditional China, young people needed to turn down their knees to show their respect to old people like their parents or their teachers. Annotations: Human={"convey", "display", "manifest"} Model= {"demonstrate"}
	As a matter of fact, young people didn't need to obey too many strict rules and could have a relatively relaxing life. Annotations: Human={"generally", "decidedly", "approximately"} Model= {"fairly"}
	To be more specific, most young men tend to be couch potatoes during weekends, leading to not concerning about social issues. Annotations: Human={"updating", "concerns", "regarding", "caring"} Model= {"worrying"}
	As a result, to make sure that I fulfil the goal I made every day, I would prevent myself from doing unnecessary things like playing for a long time in the playground as some of my classmates would do. Annotations: Human={"aim"} Model= {"objective"}
	When enjoying this piece, people can directly follow the progress of the piece to feel the picture constructed between the various instruments without exploring its creative background. Annotations: Human={"diverse"} Model= {"different"}
	No one likes to deal with people who can't adopt other people's advice and insist that only his or her idea is correct. Annotations: Human={"prefers", "enjoys"} Model= {"wants"}
	Globalists view the coronavirus as a global threat that shows the common plight of humanity and the need to work together. Annotations: Human={"reoccurring", "everyday"} Model= {"shared"}
	Second, the reading suggests that the mining industry on asteroids would be highly profitable due to the numerous valuable elements and precious metals buried under the asteroids. Annotations: Human={"implies", "proposes"} Model= {"indicates"}
OMC	Therefore, the power to resolve disputes and maintain social order is the psychological restraint of the actor and the education of the elder to the young rather than the litigation. Annotations: Human={"resolve"}, Model={"settle"}
	According to recent statistics, there is ample coverage demonstrating that humans who inhabit the hub continually will undergo severe conditions due to the dirty air and stressful surroundings. Annotations: Human ={"continually"}, Model={"constantly"}
	We are capable of making crops which can resist many kinds of unideal factors, such as colds, bugs, or even drought. Annotations: Human={resist} Model= {"withstand"}
	With the proliferation of schools and private firms nowadays, it is sometimes argued that a company or campus should have strict rules that control the type of clothing that people wear at work and at school. Annotations: Human={sometimes} Model= {"occasionally"}
	The number of students who have grades over 90 in international exams this year has surprisingly soared roughly about 10 percent, a figure that has nearly doubled as against that of last year. Annotations: Human={nearly} Model= {"almost"}
	As a result, to make sure that I fulfil the goal I made every day, I would prevent myself from doing unnecessary things like playing for a long time in the playground as some of my classmates would do. Annotations: Human={result} Model= {"consequence"}
	Even though the issue is becoming increasingly critical, many believe it is not essential for the young to learn how to plan and organize. Annotations: Human={essential} Model= {"necessary"}
	When enjoying this piece, people can directly follow the progress of the piece to feel the picture constructed between the various instruments without exploring its creative background. Annotations: Human={exploring} Model= {"examining"}
	There is no doubt that letting students decide how many and how often they could do study assignments is more flexible. Annotations: Human={often} Model= {"frequently"}
	This is because the Parliament has the intention to limit personal liberty identified as fundamental rights by the unequivocal language without allowing the diagnosed person to leave the residence or places except for certain circumstances (Momcilovic; Aboriginal Justice; s4). Annotations: Human={allowing} Model= {"permitting"}
	Those who bolster the idea might indicate this operation either is interesting to allow them to engage in a relaxed atmosphere or spend less time focusing on solid books. Annotations: Human={focusing} Model= {"concentrating"}
	What is more, if young students keep making use of video games in some lessons, their caliber of watching objects will decrease, granted that they avail themselves of exhibiting sight too much. Annotations: Human={decrease} Model= {"decline"}

Table 7: Original sentences with target token substitution in red. CD: change disagreement and OMC: only model changed.

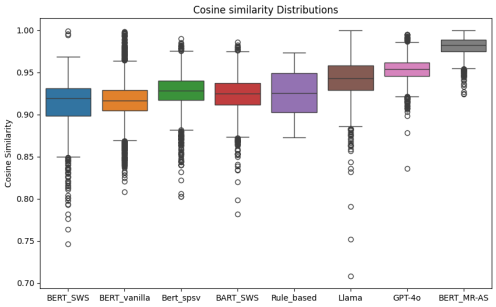

Figure 1: CS distribution results for SWS test data.

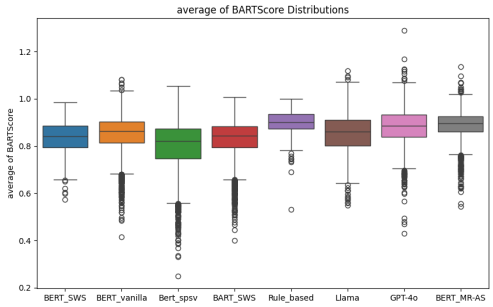

Figure 2: ABR distribution results for SWS test data.

	OAC	CA	CD	OMC	NCA
BERT-naive	**0.999**	0.888	0.339	0.274	**1**
BERT-spsv	0.994	0.981	0.583	0.486	0.999
BERT-SWS	**0.999**	0.942	0.731	0.628	**1**
BART-SWS	0.992	0.905	0.604	0.588	0.969
Rule Based	-	0.602	0.241	0.344	-
GPT-4o	-	0.761	0.563	0.661	-
LLaMa	-	0.700	0.517	0.603	-
MR+AS	**0.999**	**0.994**	**0.9**	**0.705**	**1**

Table 8: Proportion of p-values below the significance threshold ($\alpha = 0.01$) for $K_s = 1000$.

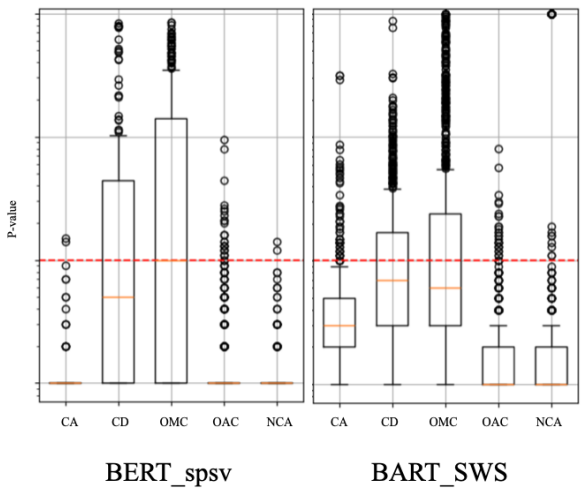

Figure 4: p-value distributions.

	OAC	CA	CD	OMC	NCA	Ratio
BERT-naive	0.914	0.032	0.054	0.052	0.948	4.09
BERT-spsv	0.952	0.02	0.028	0.017	0.983	2.82
BERT-SWS	0.738	0.113	0.149	0.064	0.936	1.65
BART-SWS	0.736	0.113	0.151	0.068	0.932	3.88
Rule Based	0.63	0.11	0.26	0.101	0.899	3.66
GPT-4o	0.508	0.244	0.248	0.115	0.885	4.28
LLaMa	0.722	0.121	0.157	0.065	0.935	4.28
MR+AS	0.893	0.055	0.052	0.027	0.973	3.96

Table 9: The value indicates the proportion of tokens involved. We divided the data into three groups (OAC, CA, CD) for changes made by human annotators, and two groups (OMC, NCA) for tokens the human annotator did not change. The ratio in the last column is between tokens both the model and the annotator change (CA and CD) and tokens only the model changed (NCA).

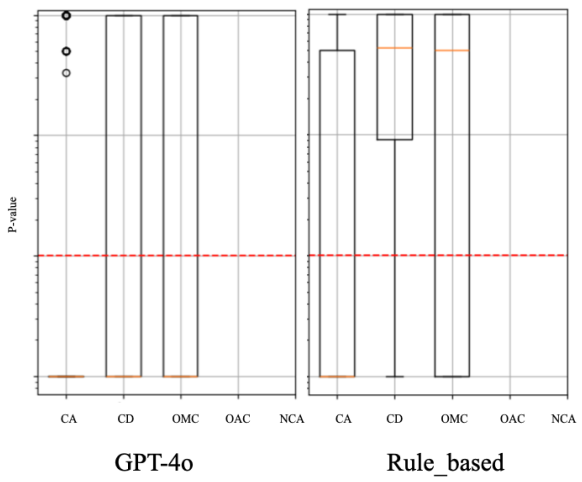

Figure 5: p-value distributions.

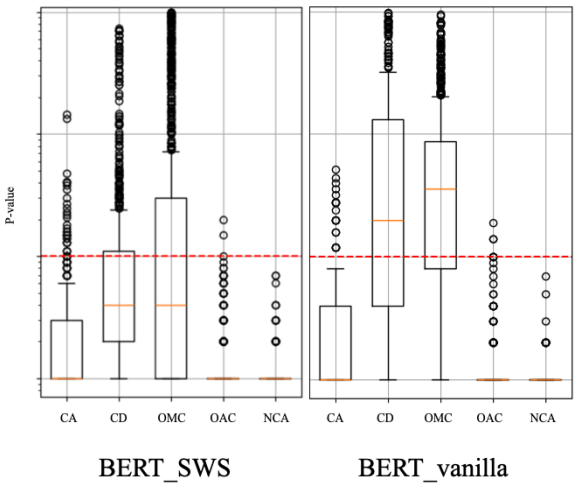

Figure 3: p-value distributions.

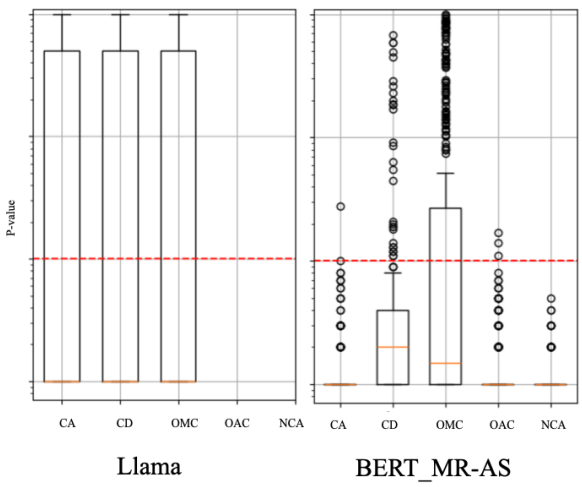

Figure 6: p-value distributions.

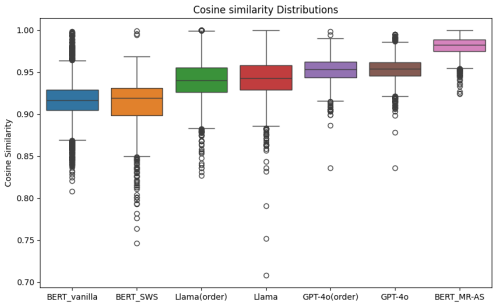

Figure 7: Results for LLaMa and GPT-4o with and without a prompt encouraging ranking.