Contextualized Semantic Distance between Highly Overlapped Texts

Letian Peng[1], **Zuchao Li**[2,*], **and Hai Zhao**[3*]
[1]Department of Computer Science and Engineering, University of California San Diego
[2]National Engineering Research Center for Multimedia Software,
School of Computer Science, Wuhan University, Wuhan, 430072, P. R. China
[3]Department of Computer Science and Engineering, Shanghai Jiao Tong University
lepeng@ucsd.edu, zcli-charlie@whu.edu.cn, zhaohai@cs.sjtu.edu.cn

Abstract

Overlapping frequently occurs in paired texts in natural language processing tasks like text editing and semantic similarity evaluation. Better evaluation of the semantic distance between the overlapped sentences benefits the language system's understanding and guides the generation. Since conventional semantic metrics are based on word representations, they are vulnerable to the disturbance of overlapped components with similar representations. This paper aims to address the issue with a mask-and-predict strategy. We take the words in the longest common sequence (LCS) as neighboring words and use masked language modeling (MLM) from pre-trained language models (PLMs) to predict the distributions in their positions. Our metric, Neighboring Distribution Divergence (NDD), represents the semantic distance by calculating the divergence between distributions in the overlapped parts. Experiments on Semantic Textual Similarity show NDD to be more sensitive to various semantic differences, especially on highly overlapped paired texts. Based on the discovery, we further implement an unsupervised and training-free method for text compression, leading to a significant improvement on the previous perplexity-based method. The high compression rate controlling ability of our method even enables NDD to outperform the supervised state-of-the-art in domain adaption by a huge margin. Further experiments on syntax and semantics analyses verify the awareness of internal sentence structures, indicating the high potential of NDD for further studies.[1]

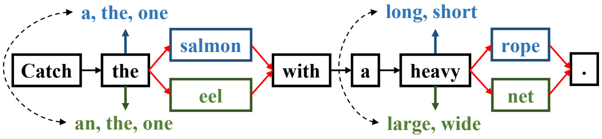


Figure 1: The comparison between two possible text scenarios with shared components. Mask-and-predicting neighboring words attenuates disturbance from overlapping when evaluating semantic distance.

1 Introduction

Comparison between highly overlapped sentences exists in many natural language processing (NLP) tasks, like text rewriting (Liu et al., 2020) and semantic textual similarity (Zhelezniak et al., 2019). A reliable evaluation of these paired sentences will benefit controllable generation and precise semantic difference understanding.

Conventional metrics, like the cosine similarity (S_C), have been popular for semantics similarity evaluation. Nevertheless, we find the evaluating capability of S_C severely degrades when the overlapping ratio rises. Niu et al. try to introduce the difference between perplexity (ΔPPL) to describe the semantic distance. Unfortunately, ΔPPL suffers from the word frequency imbalance. Also, many sentences share a similar PPL.

Based on the failure of S_C, we hypothesize that the evaluation is disturbed by the overlapped components, which share similar representations in the paired sentences. We thus intend to mitigate the disturbance and thus propose a mask-and-predict strategy to attenuate the disturbance from overlapped words. Compared to directly using the word representations for comparison, we discover that using predicted distributions from masked language modeling (MLM) results in better evaluation. Taking Figure 1 as the instance, unmasked comparison involves similar representations of *the* and *heavy* in both sentences since the encoder can see these words and encode with their information. But

[*] Corresponding author. This work was supported by Key Projects of the National Natural Science Foundation of China (U1836222 and 61733011), the Fundamental Research Funds for the Central Universities and the Fundamental Research Funds for the Central Universities (No. 2042023kf0133).

[1] Our code is released at github.com/Stareru/NeighboringDistributionDivergence/

Findings of the Association for Computational Linguistics: ACL 2023, pages 10913–10931
July 9-14, 2023 ©2023 Association for Computational Linguistics

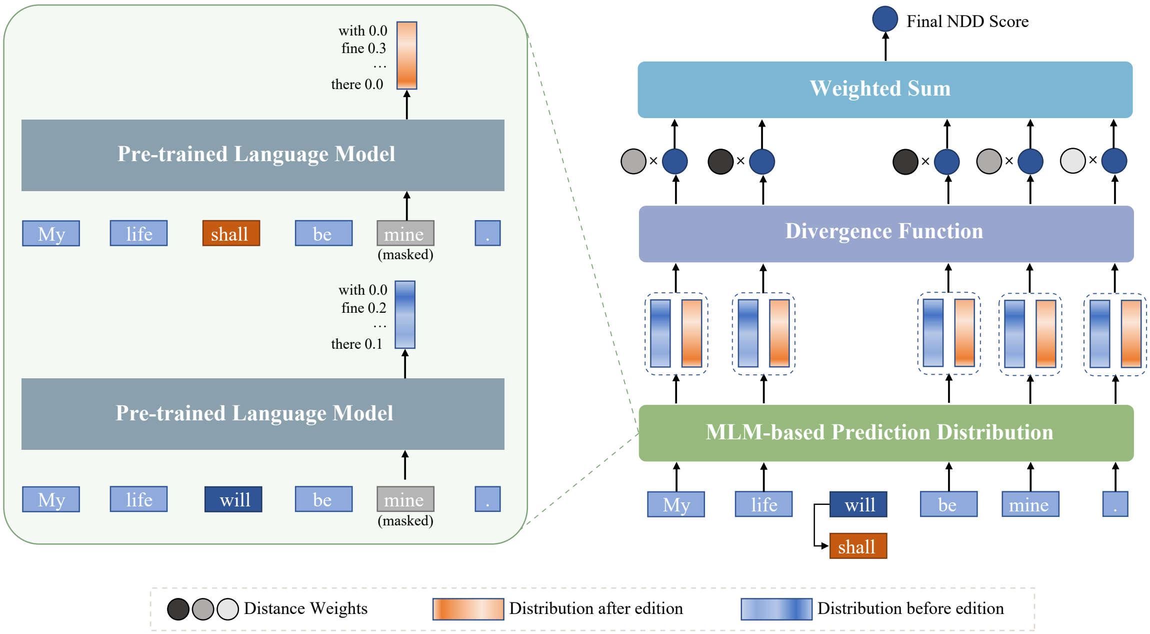


Figure 2: Calculating procedure for Neighboring Distribution Divergence.

when these words are masked, the MLM has to predict the distributions considering the contextual difference. While using representations results in a trivial *heavy*-*heavy* comparison, the difference of distributions (between candidates *long*, *short* and *large*,*wide*) better indicates how the contextual semantics changes.

Thus, we are motivated to propose a new metric, Neighboring Distribution Divergence, which compares predicted MLM distributions from pre-trained language models (PLMs) and uses the the divergence between them to represent the semantic distance. We take the overlapped words in the longest common sequence (LCS) between the paired sentences as neighboring words for the divergence calculation. We conduct experiments on semantic textual similarity and text compression. Experiment results verify NDD to be more sensitive to precise semantic differences than conventional metrics like S_C. Experiments on the Google dataset show our method outperforms the previous PPL-based baseline by around 10.0 on F1 and ROUGE scores. Moreover, the NDD-based method enjoys outstanding compression rate controlling ability, which enables it to outperform the supervised state-of-the-art by 18.8 F1 scores when adapting to Broad News Compression Corpus in a new domain. The cross-language generality of NDD is also verified by experiments on a Chinese colloquial Sentence Compression dataset.

We further use syntax and semantics analyses to test NDD's awareness of the sentence's internal structure. Our experiments show that NDD can be applied for accurate syntactic subtree pruning and semantic predicate detection. Results from our analyses verify the potential of NDD on more syntax or semantics-related tasks. Our contributions are summarized as follows:

- We address the component overlapping issue in text comparison by using a mask-and-predict strategy and proposing a new metric, Neighboring Distribution Divergence.

- We use semantic tests to verify NDD to be more sensitive to various semantic differences than previous metrics.

- NDD-based training-free algorithm has strong performance and compression rate controlling ability. The algorithm sets the new unsupervised state-of-the-art on the Google dataset and outperforms the supervised state-of-the-art by a sharp margin on the Broad News Compression dataset.

- Further syntax and semantics analyses show NDD's awareness of internal structures in sentences.

2 Neighboring Distribution Divergence

2.1 Background

Before the main discussion, we first recall the definition of perplexity and cosine similarity as the basis for further discussion.

Perplexity For a sentence with n words (more specifically, subwords) $W = [w_1, w_2, \cdots, w_n]$, perplexity refers to the average of log possibility for each word to exist in W. If the perplexity is evaluated by an MLM-based PLM, then the existing possibility is represented by the predicting distribution on the masked position.

$$W_m = [w_1, \cdots, w_{i-1}, \text{[MASK]}, w_{i+1}, \cdots, w_n]$$
$$Q = \text{PLM}^{MLM}(W_m), q_i = \text{softmax}(Q_i) \in \mathbb{R}^c$$
$$p_i = q_{i,\text{Idx}(w_i)}, \text{PPL} = \frac{1}{n}\sum_{i=0}^{n} -\log(p_i)$$

The PLM predicts the distribution Q for the masked word on i-th position. Then, the softmax function is used to get the probability distribution Q where q_j refers to the appearance possibility of j-th word in the c-word dictionary on i-th position. Here $\text{Idx}(\cdot)$ returns the index of word in the dictionary. The distribution predicting process is summarized as a function $\text{MLM}(\cdot)$ where $\text{MLM}(W, i) = q_i$.

As implausible words or structures will result in high perplexity, this metric can reflect some semantic information. Perplexity is commonly used to evaluate the plausibility of text and detect semantic errors in sentences.

Cosine Similarity For the a sentence pair W_x, W_y, a pre-trained encoder (like PLM or word embedding) encodes their contextual representations as R_x, R_y. We use PLM-based S_C for experiments and follow the best-representing scenario in (Gao et al., 2021) to use the CLS token as the sentence representation.

$$R_x = \text{PLM}(W_x), R_y = \text{PLM}(W_y)$$
$$\text{S}_C(W_x, W_y) = \frac{R_x^{CLS} \cdot R_y^{CLS}}{||R_x^{CLS}|| \times ||R_y^{CLS}||}$$

2.2 The Calculation Method

This section will detail the steps involved in determining the Neighboring Distribution Divergence. Breaking down the term NDD, **Neighboring** refers to words contained within the longest common subsequence, **Distribution** is in reference to the predicted results of the Masked Language Model on those neighboring words, while **Divergence** signifies the disparity between the predicted distributions within the LCS of the pair of sentences under scrutiny.

We'll start with a sentence pair, denoted as (W, W'). The first step is to identify the LCS between these sentences, denoted as W_{LCS}. Words within this LCS will serve as our neighboring words for comparison. The Pretraining Language Model (PLM) is applied to each word in W_{LCS} to predict their respective distributions using the MLM. Subsequently, a divergence function is employed to assess the distribution divergence between W and W' based on the same shared word. The divergence scores obtained are then assigned weights and totaled to produce the final NDD output.

The process can be mathematically expressed as:

$$q_i = \text{MLM}(W, i), q'_i = \text{MLM}(W', i)$$
$$NDD = \sum_{w \in W_{LCS}} a_w \text{F}_{div}(q_{\text{Idx}^d(w)}, q'_{\text{Idx}'^d(w)})$$

In this equation, $\text{F}_{div}(\cdot)$ symbolizes a divergence function that calculates the divergence between distributions. The functions $\text{Idx}^d(\cdot)$ and $\text{Idx}'^d(\cdot)$ are used to identify the index of w in sentences W and W' respectively. The term a_w denotes the weight assigned to each word w, which inversely corresponds to its proximity to the nearest word outside the LCS.

3 Semantic Distance Evaluation

We conduct experiments on the test dataset of Semantic Textual Similarity Benchmark [2] (STS-B) to analyze the metrics. Multiple sentence pair similarity evaluation tasks are designed to compare the metric performance and investigate the metric property.

- **Synonym-Antonym test** creates sentence pairs by replacing words with their synonyms and antonyms. Replacing by the synonym (antonym) results in a positive (negative) pair.

- **Part-of-speech (POS) test** replaces words

[2] http://ixa2.si.ehu.eus/stswiki/index.php/STSbenchmark

Metric	Syn-Ant	POS	Term	Lemma	Sup.
ΔPPL	5.3	2.8	8.7	7.9	11.2
S_C	7.8	8.1	9.1	0.0	20.8
NDD	**19.1**	**22.7**	**11.2**	**17.8**	24.0
NDD + S_C	12.5	14.5	10.8	6.8	**28.2**

Table 1: Text similarity evaluation on STS-B subset. We use Pearson Correlation as the evaluating metric. **Syn-Ant:** synonym-antonym test. **POS:** part-of-speech test. **Term:** verb term test. **Lemma:** lemma test. **Sup.:** Supervised STS test.

with ones that have the same (positive) or different (negative) parts-of-speech[3].

- **Term test** replaces verbs with ones in the same (positive) and different (negative) terms.
- **Lemma test** replaces words with ones that have the same (positive) or different (negative) lemma root.
- **Supervised test** uses the human-annotated scores for STS-B sentence pairs.

We replace 20% words for synonym-antonym, POS, and lemma tests. 100% verbs are replaced for term tests. The words for the replacement are sampled from the STS test dataset following their frequency. For the supervised test, we sample sentence pairs with an LCS that consists of at least 80% words in the shorter sentence. We use Roberta$_{base}$ as the PLM and apply Hellinger distance as the divergence function to guarantee the boundary of our metric. Mean pooling is used as the attention-assigning strategy.

$$\mathrm{H}(q, q') = \frac{1}{\sqrt{2}}\sqrt{\sum_{k=1}^{c}(\sqrt{q_k} - \sqrt{q'_k})^2} \sim [0, 1]$$

The STS experiment results are presented in Table 1. For a fair comparison, Roberta$_{base}$ is also applied to calculate S_C and PPL. NDD outperforms other metrics in all tasks, showing the strong capability of NDD to analyze semantic similarity. Also, NDD is more sensitive to POS, lemma, which is an admirable property to preserve the semantic structure for text editing.

Figure 3 shows how the ratio of overlapped words affects the metric performance. $r = 0$ indicates there is no overlapped word, so we are only able to use [CLS] and [SEP] tokens to evaluate the divergence. $r = 1$ indicates the shorter sentence is a substring of the longer sentence as all words are overlapped.

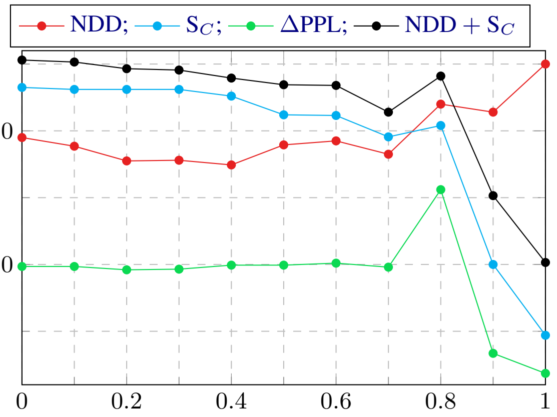


Figure 3: Relationship between metric performance on the initial test and the ratio of overlapped words. The ratio r in the x-axis means the evaluation to be on pairs where the overlapped rate of words in the shorter sentence of the pair $> x$.

While S_C performs better when fewer overlapped words hinder its evaluation, its performance severely suffers from a drop to even negative when the overlapped word ratio becomes $> 80\%$. In contrast, the rising of the ratio helps NDD perform even better as more neighboring words participate in the evaluation to provide a precise evaluation. The ensemble (ratio = $1 : 0.0025$) of NDD and S_C generally boosts the evaluating performance when the overlapped ratio $\leq 80\%$, indicating that NDD and S_C evaluate different aspects of the semantic similarity. We further discuss the metrics using specific cases in Appendix B.

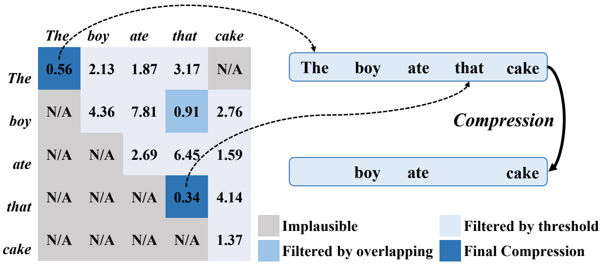
4 Unsupervised Text Compression

The prominent performance of NDD and its correlation with overlapped word ratio inspire us to apply it for extractive text compression. Text compression takes a sentence W as the input and outputs W_C where W_C is a substring of W that maintains the main semantics in W. As a substring, the compressed sentence guarantees a 100% overlapped ratio to support NDD's performance.

4.1 Span Searching and Selection

Given a sentence W, we try every span $W_{ij} = [w_i, \cdots, w_j]$ with length under a length limitation $\mathbb{L}_{max}$ for deletion. Then we use NDD to score the semantic difference caused by the deletions.

[3] Linguistic features in this paper are gotten by models from SpaCy. https://spacy.io/



	The	*boy*	*ate*	*that*	*cake*
The	0.56	2.13	1.87	3.17	N/A
boy	N/A	4.36	7.81	0.91	2.76
ate	N/A	N/A	2.69	6.45	1.59
that	N/A	N/A	N/A	0.34	4.14
cake	N/A	N/A	N/A	N/A	1.37

Figure 4: The compressing scenario of our NDD-based algorithm.

	Metric	Complexity
EVAL	ΔPPL	$2n$
	S_C	2
	NDD	$2Len(W_{LCS})$
COPRESSION	PPL Deleter	$O(n^3)$
	NDD	$O(n^3)$
	NDD w/ syn.	$O(n^2)$
	Fast NDD	$O(n^2)$
	Fast NDD w/ syn.	$O(n)$

Table 2: Time complexity of evaluating and compressing methods.

$$W'_{ij} = [w_1, \cdots, w_{i-1}, w_{j+1}, \cdots, w_n]$$
$$NDD_{ij} = \text{NDD}(W, W'_{ij})$$

As in Figure 4, We first filter W_{ij} with NDD_{ij} above the threshold $\mathbb{N}_{max}$. As overlapping still exists in searched spans, we compare each overlapped span pair and drop the span with a lower NDD score. The process iterates until no overlapped candidate exists.

4.2 Experiment

Dataset We conduct our experiments on two English datasets, Google dataset (Filippova et al., 2015) and Broadcast News Compression (BNC) Corpus[4]. On the Google dataset, we follow previous setups to use the first 1000 sentences for testing. The BNC dataset does not have a training dataset so one of the previous works (Kamigaito and Okumura, 2020) trains a compressor on the training dataset of Google for compression. We also include a Chinese colloquial Sentence Compression (SC) dataset [5] to investigate the cross-language generality of NDD. For the Chinese colloquial Sentence Compression dataset, we replaced the masks of entities with their natural language expressions[6] to avoid inaccuracy caused by them to NDD calculation.

Configuration We take cased BERT$_{base}$ as the PLM for English and BERT$_{Chinese}$ for Chinese. The divergence function is set to Kullback–Leibler (KL) divergence. The prediction on the initial text is used as the approximating distribution since it is predicted based on the text with an integral structure.

$$\text{D}_{KL}(q, q') = \sum_{k=1}^{c} q'_k \log(\frac{q'_k}{q_k}) \sim [0, \infty)$$

We fix the following hyperparameters during the experiment. $\mathbb{L}_{max}$ is set to 9 when the syntax is used and else 5. $\mathbb{N}_{max}$ is set to 1.0. Our compression is iterated for at most 5 times until no word is deleted. The weighing process can be referred to Appendix. Other parameters are adjusted to control the compression ratio. Considering the time complexity of NDD, we have developed a faster variant called Fast NDD. Fast NDD calculates the divergence by considering only the two adjacent words of the compressed span. This approach is based on the hypothesis that the nearest words are the most affected by span switching. As the effect of syntax information is shown to be effective in supervised text compression (Kamigaito and Okumura, 2020), we add a constraint that only allows dropped spans to subtrees in the syntactic dependency treebank for each step. This also boosts the efficiency as we only need to consider sparse subtree spans. The efficiency of different scenarios of NDD is shown in Table 2. Here the time complexity refers to the times of PLM-based MLM or presentation calculation. n and k refer to the length of the sentence and the dropped span, respectively.

Metric We apply the commonly-used F1 score and ROUGE metric (Lin, 2004) to evaluate the overlapping between our compression and the golden one and compare with previous works. For ROUGE, we follow the evaluating scenario in (Kamigaito and Okumura, 2020) to truncate the parts in the prediction that exceed the byte length of the golden one. We also incorporate BLEU (Papineni et al., 2002) to compare with baselines that report BLEU on the Chinese colloquial Sentence Compression dataset. Compression ratio (CR) refers to the percentage of preserved sentences in the initial sentence and $\Delta C = \text{CR}_{pred} - \text{CR}_{gold}$, which is better when being closer to 0.

[4] https://www.jamesclarke.net/research/resources

[5] https://github.com/Zikangli/SOM-NCSCM

[6] Can be found in Appendix D

	Method	F	ROUGE			CR
		F_1	R_1	R_2	R_L	CR & ΔC
	Unedited	58.2	63.8	53.4	63.3	1.00 (+0.56)
SUPERVISED	LSTM (Filippova et al., 2015)	80.0	-	-	-	0.39 (-0.05)
	LSTM-Dep‡ (Filippova et al., 2015)	81.0	-	-	-	0.38 (-0.06)
	Evaluator-SLM‡ (Zhao et al., 2018)	85.1	-	-	-	0.39 (-0.05)
	Tagger+BERT‡ (Kamigaito and Okumura, 2020)	85.0	78.1	69.9	77.9	0.40 (-0.04)
	SLAHAN‡ (Kamigaito and Okumura, 2020)	85.5	79.3	71.4	79.1	0.42 (-0.02)
UNSUPERVISED	Drop Head	31.7	23.6	14.7	22.7	0.39 (-0.05)
	Drop Tail	56.0	58.2	**47.4**	57.7	0.39 (-0.05)
	Random Drop	42.3	41.7	13.6	40.4	0.40 (-0.04)
	PPL Deleter (Niu et al., 2019)	50.0	-	-	-	0.39 (-0.05)
	PPL Deleter†	50.9	51.3	36.7	50.9	0.42 (-0.02)
	NDD (Ours)	61.2	60.3	43.2	59.6	0.41 (-0.03)
	NDD+SC‡ (Ours)	62.3	$\underline{\mathbf{62.6}}$	45.9	$\underline{\mathbf{61.9}}$	0.42 (-0.02)
	Fast NDD (Ours)	59.7	55.5	40.3	54.8	**0.43 (-0.01)**
	Fast NDD+SC‡ (Ours)	$\underline{\mathbf{67.1}}$	62.0	46.6	61.4	**0.43 (-0.01)**

Table 3: Results for sentence compression on the Google dataset. SC: Subtree Constraint with syntax treebanks. Underline: the performance improvement is significant ($p < 0.05$) considering the highest baseline. †: the method is a re-implementation. ‡: the method uses syntactic information.

Method	Training	Data	F	ROUGE			CR
			F_1	R_1	R_2	R_L	CR & ΔC
SLAHAN‡	✓	✓	57.7	40.1	30.6	39.6	0.35 (-0.36)
Fast NDD+SC‡	✗	✗	$\underline{\mathbf{76.5}}$	$\underline{\mathbf{69.8}}$	$\underline{\mathbf{55.1}}$	$\underline{\mathbf{68.5}}$	**0.70 (-0.01)**

Table 4: Comparison between the supervised state-of-the-art SLAHAN and our NDD method on BNC Corpus.

Baseline We use the PPL Deleter (Niu et al., 2019) as the main baseline. Deleter uses ΔPPL to control the compressing procedure and tries to preserve a lower PPL in each step. Simple baselines that directly drop words according to the compression ratio are also included. We report several supervised results to show the current development on the tasks.

Google Table 3 presents our results on the Google dataset. Compared to the PPL Deleter, the basic NDD leads to a sharp improvement on all metrics, 10.3 improvement on the F1 score, and 8.7 on ROUGE$_L$. Fast NDD underperforms the initial NDD, but the performance is admirable considering its efficiency. Our method benefits from syntactic constraints, especially for Fast NDD. Syntactic constraints boost Fast NDD's performance to around 7.0 on most metrics to set the new unsupervised state-of-the-art on F1 score. Still, the initial NDD method with syntax is state-of-the-art on ROUGE metrics. Thanks to the compression rate controlling ability of our method, we can control the compression to a CR extremely close to the golden one.

BNC The BNC Corpus is a perfect case to show the advantage of NDD's ability to control the compression rate. We take the supervised SOTA syntactically look-ahead attention network (SLAHAN) (Kamigaito and Okumura, 2020) as the baseline. Since BNC does not have a training dataset, SLAHAN is trained on the $200K$ Google corpus. Nevertheless, the cross-domain adaption of SLAHAN is not successful as its ΔC is an extremely negative -0.35 in Table 4. In contrast, our PLM-based unsupervised method enjoys robustness and can be easily adapted to different domains, and reach a CR close to the golden one. Our unsupervised method thus outperforms the supervised state-of-the-art by a huge margin (20 $\sim$ 30) on all metrics.

Colloquial SC The experimental results depicted in Table 5 underline the cross-lingual generality of our NDD method. Notably, our method surpasses the PPL Deleter in performance, thereby setting a new benchmark for unsupervised models across all evaluation metrics. The compression rate controlling ability of NDD further allows it to generate BLEU scores that are in close alignment with the supervised Tagger+BERT model, indicating

	Method	F	BLEU				ROUGE			CR
		F_1	B_1	B_2	B_3	B_4	R_1	R_2	R_L	CR & ΔC
	Unedited	81.7	74.4	73.7	71.4	68.5	70.4	55.3	70.2	1.00 (+0.32)
SUP.	Tagger+BERT (Zi et al., 2021)	85.8	73.4	64.3	58.3	53.9	-	-	-	0.83 (+0.15)
	SOM-NCSCM (Zi et al., 2021)	89.7	84.0	78.2	74.8	70.1	-	-	-	**0.68 (-0.00)**
UNSUPERVISED	PPL Deleter†	56.3	51.8	47.3	43.5	40.5	55.7	25.3	55.4	0.70 (+0.02)
	NDD (Ours)	74.8	66.4	57.1	50.5	44.6	67.8	47.0	67.8	0.65 (-0.03)
	NDD+SC‡ (Ours)	76.4	68.5	59.3	52.8	47.2	69.3	49.9	69.2	**0.68 (-0.00)**
	NDD (wwm)+SC‡ (Ours)	**76.7**	**68.6**	**59.6**	**53.5**	**47.9**	**70.5**	**50.8**	**70.2**	0.70 (+0.02)
	Fast NDD (Ours)	73.9	64.1	56.6	51.2	45.6	65.9	48.4	65.7	0.67 (-0.02)
	Fast NDD+SC‡ (Ours)	75.5	66.9	58.5	52.4	47.2	67.2	47.7	66.8	0.69 (+0.01)
	Fast NDD (wwm)+SC‡ (Ours)	74.7	66.0	57.7	51.7	45.7	67.7	48.4	67.5	0.70 (+0.02)

Table 5: Results for sentence compression on the Chinese colloquial Sentence Compression dataset.

$\mathbb{N}_{max}/\mathbb{L}_{max}$	F	ROUGE			CR
	F_1	R_1	R_2	R_L	CR & ΔC
4.00 / 5	44.6	35.8	15.4	35.7	0.16 (-0.28)
2.00 / 5	58.5	52.4	34.9	52.1	0.27 (-0.17)
1.00 / 5	67.1	62.0	46.6	61.4	**0.43 (-0.01)**
0.50 / 5	**67.7**	64.6	51.2	64.1	0.57 (+0.13)
0.25 / 5	66.0	**65.1**	**53.0**	**64.6**	0.69 (+0.25)
1.00 / 2	66.8	63.0	47.5	62.4	0.48 (+0.04)
1.00 / 1	64.7	64.2	49.3	63.7	0.62 (+0.18)

Table 6: Performance results of different configuration setups on the Google dataset.

the strength of our approach. In our experiments, we also deployed a whole-word-masking (wwm) Roberta[7] as the PLM. This led to additional performance enhancements, which indicates the accuracy of NDD can benefit from whole-word-masking during the pre-training.

In summary, our NDD method coupled with the Subtree Constraint offers the best overall performance among unsupervised models. It achieves the highest F_1 score of 76.7, surpasses all others in most metrics, and is very close to the best CR. This confirms its strong potential for the task of sentence compression across languages.

Compression Rate Controlling We provide a more specific analysis of NDD's compression rate controlling ability. By changing the configuration of our scenario, our method can result in different compression ratios, from 16% to 69%. When the compression ratio is higher than 43%, NDD always results in text with admirable quality (F1 $> 60\%$, ROUGE$_{1\&L}$ $> 60\%$). Also, when the CR is extremely small, NDD can still preserve much information in the initial sentence, with overlapping F1 score 58.5 for 27% and 44.6 for 16%. Also, adjusting the compressing iteration for the same configuration setup can result in high-quality output in different compression ratios. The compression rate controlling ability enables our method to easily adapt to systems requiring different compression ratios. Further case-based discussion can be referred to Appendix H.

5 Further Analysis[8]

We continue studying the compression algorithm to further investigate NDD's syntax awareness via analyzing the roles of pruned words in the syntax treebank.

5.1 Syntax Subtree Pruning

This task tests whether NDD is able to detect syntactic structures using syntax treebanks. (1) If the pruned nodes mostly play subordinated roles in the tree, our algorithm can be better certificated to compress with an awareness of syntax. We depict an instance of syntax treebank in Figure 5. In the treebank, deeper nodes like *the* and *that* are less important for the integrity of syntax structure. (2) Also, pruning a subtree like *that cake* will preserve more syntax structure than pruning a non-subtree like *ate that*. Thus, we introduce two metrics to evaluate the pruning performance: **Depth-n** and **Subtree-k**.

$$\text{Depth-}n = \frac{Count(w|Depth(w) = n)}{Count(w)}$$

$$\text{Subtree-}k = \frac{Count(s|IsSub(s), Len(s) = k)}{Count(s|Len(s) = k)}$$

[7] https://huggingface.co/hfl/chinese-roberta-wwm-ext

[8] In analyses, we continue using the KL divergence as the divergence function.

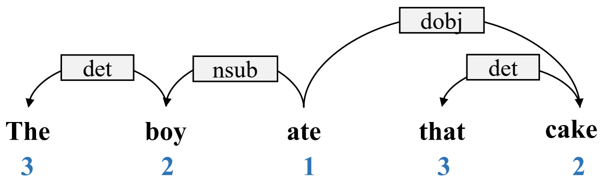


Figure 5: Nodes and their depths (blue) in a syntax treebank. Deeper nodes in the treebank generally play less important roles in context.

Method	$\mathbb{N}_{max}$	**Depth-n**				**Subtree-k**		
		1	2	3	≥ 4	1	2	≥ 3
Random	1.0	4	21	22	54	55	31	27
	2.0	4	24	23	48	52	29	23
PPL	1.0	3	20	21	56	79	57	48
	2.0	3	23	24	50	71	45	37
NDD	1.0	1	19	20	60	90	81	66
	2.0	2	23	23	52	82	70	63

Table 7: Proportion (%) of pruned nodes in certain depths of the syntax treebanks and proportion (%) of pruned spans that are subtrees. $\mathbb{L}_{max}$ is set to 5.

where w, s represent the pruned words and spans. $Count(\cdot)$ returns the number of items, $Depth(\cdot)$ returns the depth of a word in the syntax treebank, $IsSub(\cdot)$ return if a span is a subtree in the treebank, and $Len(\cdot)$ returns the number of words in a span. Depth-n and Subtree-k thus reflect the word-level and span-level pruning quality, respectively.

We experiment on the PTB-3.0 test dataset (Marcus et al., 1993). We use the random dropping strategy with the same compression ratio as the baseline for comparison. As in Table 7, the proportion of nodes in shallower levels (depth=1 $\sim$ 3) pruned by our algorithm is smaller than all the corresponding random and PPL-based pruning. Also, the proportion of subtrees in spans pruned by the NDD-based algorithm is significantly larger than in other correspondents. Thus, we conclude that NDD can guide the compressing algorithm to detect subordinated components in syntax dependency treebanks.

5.2 Predicate Detection

To explore the semantics awareness of NDD, we experiment on the semantic role labeling (SRL) task for predicate detection. As predicates are semantically related to more components (augments) in sentences, deleting them or replacing them with stop words will result in a larger semantic distance from the initial sentence. We evaluate the predicate detecting ability following the words ranking task.

Edit	ENG-ID		ENG-OOD		SPA	
	mAP	AUC	mAP	AUC	mAP	AUC
(PPL-based)						
Delete	36.8	56.8	44.5	60.4	26.6	54.5
Mask Replace	35.9	56.7	33.1	48.5	25.1	50.4
(NDD-based)						
Delete	53.1	77.0	62.5	80.8	48.8	77.1
Mask Replace	48.0	74.4	57.3	80.6	44.2	75.0
Stop Word Replace	49.9	76.4	56.4	78.5	45.2	77.2
Ensembled	**54.3**	**79.7**	**63.1**	**83.3**	**54.7**	**82.8**

Table 8: Evaluation on ability of metrics to detect predicates in sentences.

We rank the probability of words to be predicates according to NDD evaluation and evaluate the detecting performance by ranking metrics: mean average precision (mAP) and area under curve (AUC).

We conduct our experiments on Conll-2009 SRL datasets[9] (Hajic et al., 2009). To test our method‘s generality, in-domain (ID) and out-of-domain (OOD) English (ENG) datasets are included. Another Spanish (SPA) dataset is also used for cross-language evaluation. To generate a new sentence for semantic distance computation, we edit each word in the sentence in three ways: (a) Deletion, (b) Replacement with a mask token, (c) Replacement with a stop word[10]. We apply cased SpanBERT$_{base}$ (Joshi et al., 2020) and cased BERT$_{Spanish}$[11] (Cañete et al., 2020) as PLMs. For comparison, we implement a PPL-based algorithm that uses ΔPPL to detect predicates.

Our results are presented in Table 8. The generally poor performance shows that ΔPPL might not be a proper metric for predicate detection. In contrast, the NDD-based algorithm produces much better results and outperforms the PPL-based algorithm by 10 $\sim$ 20 scores on both AUC and mAP metrics, which is a remarkably significant margin and verifies NDD to be much more capable in understanding semantics. The ensemble of three processes boosts AUC, mAP to higher than 80.0, 50.0, respectively, making it a plausible way to detect predicates following an unsupervised procedure.

6 Related Works

The evaluation on text similarity provides valuable guidance on various downstream tasks, including text classification (Park et al., 2020), document

[9] Instances with length ≤ 50, number of predicates > 0.

[10] *a* for ENG-ID, *that* for ENG-OOD and *el* for SPA

[11] https://huggingface.co/dccuchile/bert-base-spanish-wwm-cased

clustering (Lakshmi and Baskar, 2021), and translated text detection (Nguyen-Son et al., 2021). The commonly used cosine similarity evaluates paired sentences' similarity based on the cosine value between word embeddings or pre-trained representations (Reimers and Gurevych, 2019; Zhang et al., 2020b). Unfortunately, when the overlapping ratio between paired sentences rises, the representation-based method suffers from faults caused by similar word representations. Our work replaces word representations with predicted distributions to mitigate the disturbance from overlapped components.

The proposal of PLMs (Devlin et al., 2019) inspires researchers to leverage the upstream training process for text similarity evaluation. Niu et al. leverage the perplexity calculated from PLMs to represent the semantic distance between texts during text compression. While perplexity can evaluate the fluency of sentences, a recent study (Kuribayashi et al., 2021) suggests that low perplexity does not directly refer to a human-like sentence. Also, perplexity fails with words that share a similar existing probability but are with opposite or irrelevant meanings. Other PLM-based metrics like BERTScore have been verified by experiments to evaluate text generation better (Zhang et al., 2020a). Other pre-trained models for evaluation are also an interesting topic. To evaluate semantics preservation in AMR-to-sentence, Opitz and Frank exploits AMR parser to compare the AMR graph of generated results with the golden graph, showing the potential of pre-trained models to evaluate more complex linguistic structures.

Many supervised methods (Malireddy et al., 2020; Nóbrega et al., 2020) have been proposed for text compression. Syntax treebanks play a critical role in text compression (Xu and Durrett, 2019; Wang and Chen, 2019; Kamigaito and Okumura, 2020). Unsupervised methods have been explored to extract sentences from documents to represent key points (Jang and Kang, 2021). Nevertheless, span pruning is still far from satisfaction. As mentioned before, (Niu et al., 2019) explores using ΔPPL for compression, which is not so capable as NDD in semantics preservation.

Syntax and semantic analyses (Dozat and Manning, 2017; Li et al., 2020b,a,c) reflect model's awareness of the internal structures in sentences. The awareness of syntax and semantics of NDD is verified by those tasks.

7 Conclusion

We address the overlapping issue in semantic distance evaluation in this paper. To mitigate the disturbance from overlapped components, we mask and predict words in the LCS via PLM-based MLM. NDD evaluates the semantic distance using a weighted sum of the divergence between predicted distributions. STS experiments verify NDD to be more sensitive to a wide range of semantic differences and perform better on highly overlapped paired texts, which is challenging for conventional metrics. NDD-based text compression algorithm significantly boosts the unsupervised performance, and its high compression rate controlling ability enables the adaption to datasets in different domains. NDD's awareness of syntax and semantics is verified by further analyses, showing the potential of NDD for further studies.

Limitations

While our NDD metric has demonstrated its effectiveness in measuring the semantic distance between overlapped sentences, there are still some limitations to consider. Firstly, the calculation efficiency of NDD may become a bottleneck when dealing with large amounts of data. The mask-and-predict strategy requires the generation of a large number of predictions for each word in the LCS, which can be computationally expensive. Therefore, for large-scale applications, more efficient algorithms or hardware acceleration may be necessary to speed up the calculation of NDD. Secondly, our method currently cannot selectively compress certain parts of the text. The mask-and-predict strategy compresses the entire overlapped segment, which may not always be desirable. For example, in some cases, it may be more desirable to compress only the less relevant portion of the text while retaining the most informative content. While NDD has an advantage over supervised compressors in controlling compression ratio, it still cannot control the compression orders. Future research may investigate techniques to allow for more fine-grained control over the compression process. Overall, while NDD shows great promise in improving the evaluation of semantic similarity and text compression, further research is needed to address these limitations and improve the compression rate controlling ability and versatility of the method.

References

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Katja Filippova, Enrique Alfonseca, Carlos A. Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 360–368. The Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics.

Jan Hajic, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Stepánek, Pavel Stranák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2009, Boulder, Colorado, USA, June 4, 2009*, pages 1–18. ACL.

Myeongjun Jang and Pilsung Kang. 2021. Learning-free unsupervised extractive summarization model. *IEEE Access*, 9:14358–14368.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguistics*, 8:64–77.

Hidetaka Kamigaito and Manabu Okumura. 2020. Syntactically look-ahead attention network for sentence compression. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8050–8057. AAAI Press.

Tatsuki Kuribayashi, Yohei Oseki, Takumi Ito, Ryo Yoshida, Masayuki Asahara, and Kentaro Inui. 2021. Lower perplexity is not always human-like. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5203–5217. Association for Computational Linguistics.

R. Lakshmi and S. Baskar. 2021. Efficient text document clustering with new similarity measures. *Int. J. Bus. Intell. Data Min.*, 18(1):49–72.

Tao Li, Parth Anand Jawale, Martha Palmer, and Vivek Srikumar. 2020a. Structured tuning for semantic role labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8402–8412. Association for Computational Linguistics.

Zuchao Li, Hai Zhao, and Kevin Parnow. 2020b. Global greedy dependency parsing. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8319–8326. AAAI Press.

Zuchao Li, Hai Zhao, Rui Wang, and Kevin Parnow. 2020c. High-order semantic role labeling. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1134–1151. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Qian Liu, Bei Chen, Jian-Guang Lou, Bin Zhou, and Dongmei Zhang. 2020. Incomplete utterance rewriting as semantic segmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2846–2857. Association for Computational Linguistics.

Chanakya Malireddy, Tirth Maniar, and Manish Shrivastava. 2020. SCAR: sentence compression using autoencoders for reconstruction. In *Proceedings of* *the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, ACL 2020, Online, July 5-10, 2020*, pages 88–94. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguistics*, 19(2):313–330.

Hoang-Quoc Nguyen-Son, Tran Thao Phuong, Seira Hidano, Ishita Gupta, and Shinsaku Kiyomoto. 2021. Machine translated text detection through text similarity with round-trip translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5792–5797. Association for Computational Linguistics.

Tong Niu, Caiming Xiong, and Richard Socher. 2019. Deleter: Leveraging BERT to perform unsupervised successive text compression. *CoRR*, abs/1909.03223.

Fernando Antônio Asevedo Nóbrega, Alípio M. Jorge, Pavel Brazdil, and Thiago A. S. Pardo. 2020. Sentence compression for portuguese. In *Computational Processing of the Portuguese Language - 14th International Conference, PROPOR 2020, Evora, Portugal, March 2-4, 2020, Proceedings*, volume 12037 of *Lecture Notes in Computer Science*, pages 270–280. Springer.

Juri Opitz and Anette Frank. 2021. Towards a decomposable metric for explainable evaluation of text generation from AMR. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 1504–1518. Association for Computational Linguistics.

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.

Kwang-Il Park, June Seok Hong, and Wooju Kim. 2020. A methodology combining cosine similarity with classifier for text classification. *Appl. Artif. Intell.*, 34(5):396–411.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Yifan Wang and Guang Chen. 2019. Improving a syntactic graph convolution network for sentence compression. In *Chinese Computational Linguistics - 18th China National Conference, CCL 2019, Kunming, China, October 18-20, 2019, Proceedings*, volume 11856 of *Lecture Notes in Computer Science*, pages 131–142. Springer.

Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression. 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 3292–3303, Hong Kong, China. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020a. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2020b. An unsupervised sentence embedding method by mutual information maximization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1601–1610, Online. Association for Computational Linguistics.

Yang Zhao, Zhiyuan Luo, and Akiko Aizawa. 2018. A language model based evaluator for sentence compression. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 170–175. Association for Computational Linguistics.

Vitalii Zhelezniak, Aleksandar Savkov, April Shen, and Nils Y. Hammerla. 2019. Correlation coefficients and semantic textual similarity. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 951–962. Association for Computational Linguistics.

Kangli Zi, Shi Wang, Yu Liu, Jicun Li, Yanan Cao, and Cungen Cao. 2021. SOM-NCSCM : An efficient neural chinese sentence compression model enhanced with self-organizing map. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 403–415. Association for Computational Linguistics.

A Dataset Statistics

Dataset	Inst. Num.	Avg. Len.	CR
Google	1000	27.71	0.44
BNC	595	31.15	0.71
Colloquial SC	150	8.13	0.68
STS-B	2758	9.81	-
PTB	2416	23.46	-
Conll09-ENG-ID	2399	24.04	-
Conll09-ENG-OOD	425	16.96	-
Conll09-SPA	1725	29.35	-

Table 9: Statistics of our datasets in experiments.

B Specific Cases for Semantic Difference Evaluation

We use specific cases to further explore the ability of NDD to capture precise semantic differences using several examples. As in Table 10, we edit the initial sentence *"I am walking in the cold rain."* with a series of replacements. We keep the syntactic structure of the sentence unchanged and replace some words with other words of the same part-of-speech. Thus, the difference between the initial and edited sentences is majorly the semantics.

Sentence	PPL	NDD	S_{C-}
I am walking in the cold rain.	5.99	0.00	1.000
I am walking in the cool rain.	10.10	0.81	0.995
I am walking in the freezing rain.	5.63	0.97	0.997
I am walking in the heavy rain.	5.30	1.82	0.994
I am walking in the hot rain.	14.77	3.17	0.995
I am walking in the cold snow.	5.37	2.46	0.996
I am walking in the cold night.	6.18	3.52	0.991
I am walking in the cold sunshine.	8.59	4.73	0.994
I am running in the cold rain.	11.86	0.66	0.990
I am wandering in the cold rain.	16.89	0.89	0.982
I am swimming in the cold rain.	14.84	3.29	0.986
I was walking in the cold rain.	10.32	4.72	0.980
He am walking in the cold rain.	105.55	13.04	0.991
He is walking in the cold rain.	13.95	7.22	0.980

Table 10: Cases for detection of NDD on very precise semantic difference. The initial sentence is *"I am walking in the cold rain."*

We divide the editing cases into several groups. In the first three groups, we change words (adjective, noun, and verb respectively) into similar, different, or opposite meanings. NDD successfully detects the semantic difference and precisely evaluates changing extents. Taking the first group as an instance, changing from *cold* into *cool* and *freezing* keeps most semantics while changing into *hot* leads to the opposite and even implausible semantics. NDD reflects the difference of semantics between these edited results and assigns a much higher score to the *cold*-to-*hot* case. Moreover, in the medium case where the aspect for description is changed to *heavy*, NDD remarkably assigns a medium score to this case, showing its high discerning capability.

In the last case group, we change the tense and subject of the sentence. NDD is shown to be fairly sensitive to tenses and subjects. This property can be used to retain those critical properties during edits. NDD is also able to detect syntactic faults like the combination of *He am* and can thus be used for fault prevention during the edit.

From these cases, we can also see why perplexity and cosine similarity is incapable of detecting precise semantic difference as NDD. In Table 10, cosine similarity cannot detect the subtle semantic difference and even syntactic faults. We attribute this to the high reliance on word representations for sentence representations, as sentences with many words overlapped will be classified to be similar.

For perplexity (PPL), the first problem with it is that this metric evaluates the fluency of a single sentence. Perplexity will thus guide edits to transform sentences into more syntactically plausible versions, ignoring semantics. As a result, edited results with lower perplexity may change semantics like *cold*-to-*heavy* and *rain*-to-*snow*. NDD is able to preserve semantics much better by suggesting changing *cold* to *cool* or *freezing* and changing *walking* to *running* or *wandering*.

Another reason is that perplexity can easily be misguided by low-frequency words. In the *walking*-to-*wandering* case, since *wandering* is a low-frequency word, the resulted perplexity is even higher than the *walking*-to-*swimming* case. Since perplexity is scored based on the existence probability of words, the low-frequency *wandering* will lead to a higher perplexity, even though *wandering* is semantically closer to *walking* than *swimming*. This issue is overcome in NDD as we use predicted distributions rather than real words. As described before, NDD can understand low-frequency words and even named entities much better. As a result, NDD correctly scores the semantic difference caused by replacement on *walking*.

C Other Details for Compression

For weighing in text compression, we modify the exponential weight and use the balanced weights

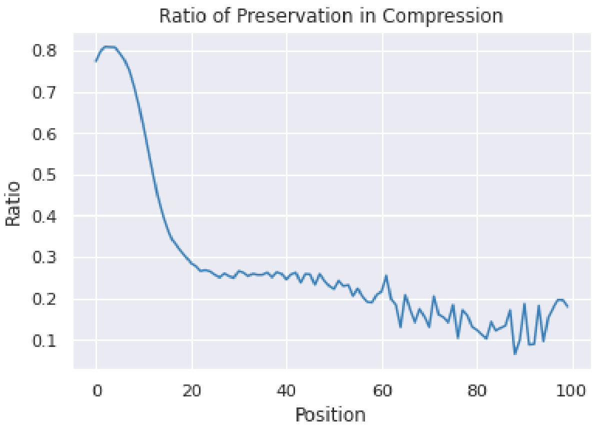


Figure 6: The ratio of preserved tokens in certain positions of the initial sentence. Statistics from the Google training dataset.

for distance.

$$
\begin{aligned}
a_k &= \mu^{min(|k-i|,|k-j|)} \\
a'_k &= a_k + a_{n'-k} * \mu^{n'} \\
n' &= n - (j - i + 1)
\end{aligned}
$$

where n is the length of the initial sentence, k is the neighboring word's position, and i, j are the start and end positions of the pruned span. The modification guarantees the total distance weights are the same for each NDD calculation, while the exponential weight assigns fewer weights to words on two sides of the sentence.

Furthermore, we add another weight b_k to encourage our algorithm to delete later words in the sentence. As shown in Figure 6, later words are less common to be used for summary. We modify the weighted sum as follows.

$$
\begin{aligned}
b_w &= \nu^{Idx(w)} \\
NDD &= \sum_{w \in W_{LCS}} a'_w b'_w \mathrm{F}_{div}(q_{\mathrm{Idx}^d(w)}, q'_{\mathrm{Idx}'^d(w)})
\end{aligned}
$$

In experiments, we fix μ to 0.9 and adjust ν to adapt to the compression rate.

Method	F	ROUGE			CR
	F$_1$	R$_1$	R$_2$	R$_L$	CR & ΔC
PPL Deleter	50.9	51.3	36.7	50.9	0.42 (-0.02)
PPL Deleter+SC‡	53.1	54.7	40.3	54.5	0.42 (-0.02)
S$_C$	46.5	45.0	16.9	43.9	0.37 (-0.07)
S$_C$+SC‡	49.5	50.5	23.0	49.7	0.42 (-0.02)
BERTScore	48.0	48.6	15.8	47.4	0.41 (-0.03)
BERTScore+SC‡	47.2	49.5	18.8	48.6	0.39 (-0.05)
NDD*	60.1	59.8	41.7	59.2	0.41 (-0.03)
NDD+SC*‡	60.8	61.9	44.5	61.3	**0.45 (+0.01)**
NDD	61.2	60.3	43.2	59.6	0.41 (-0.03)
NDD+SC‡	62.3	**62.6**	45.9	**61.9**	0.42 (-0.02)
Fast NDD	59.7	55.5	40.3	54.8	**0.43 (-0.01)**
Fast NDD+SC‡	**67.1**	62.0	**46.6**	61.4	**0.43 (-0.01)**

Table 12: Extra comparison including S$_C$ and BERTScore. $*$: use $kl(d||d')$ instead of $kl(d'||d)$

D Mask to Expression

Mask	Expression
[业务名词]	某业务
[电话号码]	电话
[编号]	1
[名词]	这个
[地址]	某地

Table 11: The dictionary that transforms Chinese masks to natural language expressions.

E Extra Comparison

We further investigate the capability difference of different metrics in text compression. As in Table 12, we replace the evaluator in the compressing scenario with S$_C$ and BERTScore (Zhang et al., 2020a). The experiment results show a large gap between NDD and other metrics, verifying the prominent semantic distance evaluating the capability of NDD.

F Human Evaluation

We further use human evaluation to compare the performance of text compression algorithms. We sample 100 sentences from the Google test dataset and ask human evaluators to score for the syntactic and semantic integrity of the output. The evaluators are blind to which algorithm compresses and produces the output. We assign scores from 0 to 5 as follows:

- 0: No legal structure, totally a combination of meaningless fragments.

Method	Syntax	Semantics
PPL Deleter	3.69	2.56
PPL Deleter+SC‡	3.93	2.88
S_C	1.87	1.51
S_C+SC‡	2.35	2.08
BERTScore	1.96	1.83
BERTScore+SC‡	2.38	2.17
NDD	3.87	3.46
NDD+SC‡	**4.08**	**3.62**
Fast NDD	3.73	3.18
Fast NDD+SC‡	4.01	3.43

Table 13: Human evaluation on the syntax and semantics integrity of outputs from unsupervised text compression algorithms.

- 1: Poor structure, only some meaningful components, and the whole structure are not understandable.
- 2: The whole structure is acceptable but contains faults compared to the initial sentence.
- 3: Some parts of the initial structure are preserved, but the compression drops some important components.
- 4: Most parts of the initial structure are preserved, still there exists a little inconsistency or ignorance of important components.
- 5: The structure is as integral as human's.

The human evaluation verifies NDD to keep a large gap with conventional metrics in text compression in syntactic and semantic integrity. Also, the benefit of introducing syntactic constraints is shown in every algorithm.

G NDD distributions

To more specifically present how NDD is sensitive to semantic differences, we depict the distribution of bounded (Hellinger distance-based) and unbounded (KL divergence-based) NDD in Figures 7 and 8.

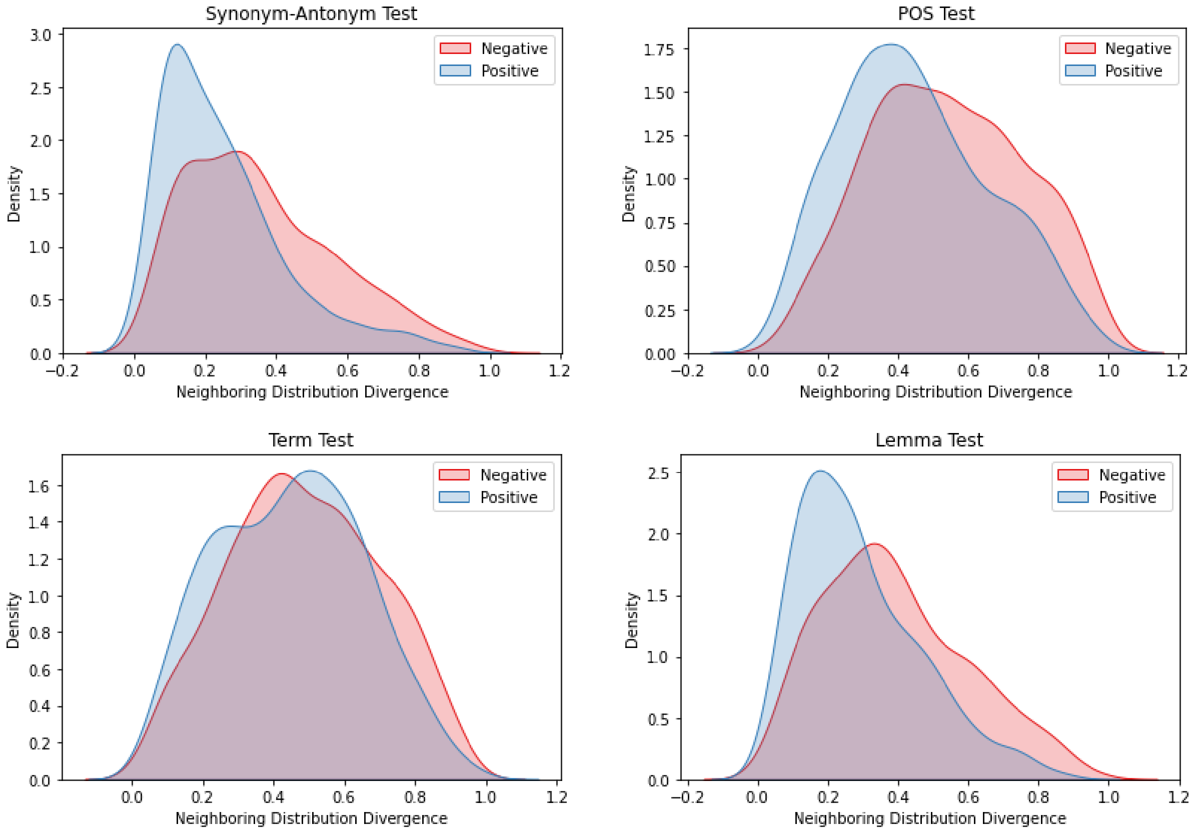


Figure 7: Distribution of bounded NDD (Hellinger distance) on semantic difference tests.

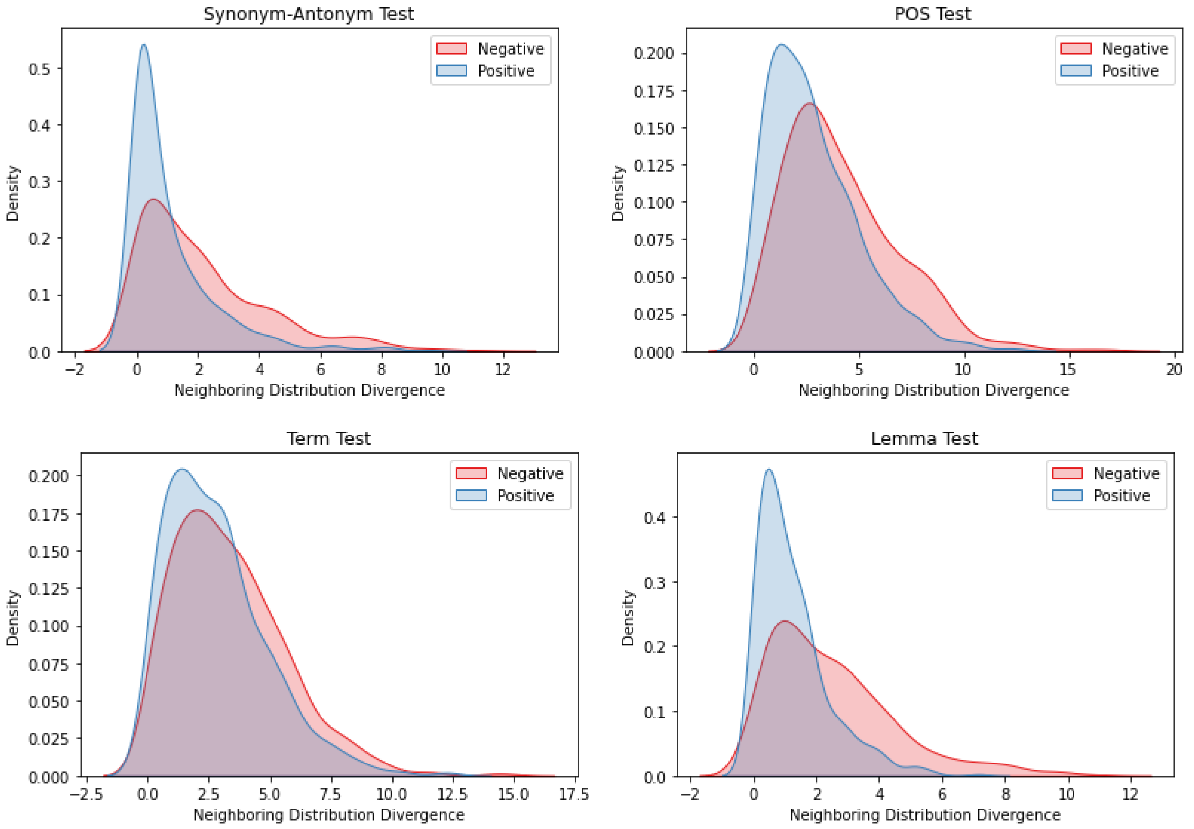


Figure 8: Distribution of unbounded NDD (KL divergence) on semantic difference tests.

H Compression Cases

Init: The speed limit on rural interstate highways in Illinois will be raised to 70 mph next year after Gov. Pat Quinn approved legislation Aug. 19, despite opposition from the Illinois Dept. of Transportation, state police and leading roadway safety organizations. **Edit:** The speed limit will be 70 mph despite opposition from organizations. **Gold:** The speed limit on highways in Illinois will be raised to 70 mph next year. **F1 Score =** 51.9 (↓ 8.5) **ROUGE =** 53.8
Init: New US ambassador to Lebanon David Hale presents credentials to Lebanese President Michel Sleiman in Baabda, Friday, Sept. 6, 2013. **Edit:** New US ambassador to Lebanon presents credentials to Lebanese President Michel Sleiman. **Gold:** New US ambassador presents credentials to Michel Sleiman. **F1 Score =** 87.0 (↑ 28.7) **ROUGE =** 75.0

Table 14: Examples for how automatic metrics reflect the performance of NDD-based compression. Improvement refers to comparison with unedited texts.

Real Effect v.s. Automatic Metrics As the compressed results for sentences can be various, automatic metrics might not be able to fully reflect the compressing ability of our algorithm. Also, as our compression follows a training-free procedure, the compressed results might not be in the same style as the annotated golden ones like the first instances in Table 14. Both our compressed and the golden result keep the main point that *the speed limit will be 70 mphs*, preserving the semantics of the whole sentence. Nevertheless, the golden compression tends to keep some auxiliary information like the location *on highways in Illinois* and the time *next year*. In contrast, NDD-based compression tends to remove that unimportant information and prevent semantics in other parts of the sentence from being unchanged. Thus, NDD-based compression still keeps *despite opposition from organizations* towards the integrated semantics. In the second instance of Table 14, as the golden compression also removes location and time information from the sentence, our algorithm leads to a significant improvement since our compressing style matches with the annotated one. Considering that the automatic metrics may be biased due to the style of annotation, we present more cases in this section to show the capacity of our algorithm to keep semantics and fluency while removing unimportant and auxiliary components at the same time.

Init: A US$5 million fish feed mill with an installed capacity of 24,000 metric tonnes has been inaugurated at Prampram, near Tema, to help boost the aquaculture sector of the country.
Iter1: **A** US$5 million **fish feed mill with** an installed **capacity** of **24,000** metric tonnes **has been inaugurated at Prampram**, near Tema, **to** help **boost the aquaculture sector** of the country**.**
Iter2: **A** fish feed **mill** with capacity 24,000 **has been inaugurated** at Prampram **to boost** the **aquaculture sector.**
Final: A mill has been inaugurated to boost aquaculture sector.

Table 15: Cases for output in iterations of the NDD-based compression. **Bold: Kept components**

Outputs from Compression Iterations We present the intermediate outputs of our algorithm in Table 15. NDD-based text compression is shown to be capable of detecting and removing auxiliary components like locations or adjective spans in the sentence, for example. Also, the syntactic integrity and initial semantics are preserved in each iteration of our algorithm. There is an advantage over supervised methods as output in each iteration is still a plausible compression for the initial sentence. We can thus set some proper thresholds and iterate the compression until we get a fully satisfying output.

Init: 调价周期内，沙特下调10月售往亚洲的原油价格，我国计划释放储备原油，油价一度承压下跌。 (Translation) During the price adjustment, Saudi scales down the price of crude oil sold to Asia in October, our country plans to release the reserved crude oil, oil price has once been under the dropping pressure. **Edit:** 调价周期内，沙特下调原油价格，我国释放储备原油。 (Translation) During the price adjustment, Saudi scales down the price of crude oil, our country releases the reserved crude oil.
Init: El comité de crisis, aseguró el presidente, ha tomado decisiones estratégicas que, por seguridad, no pueden ser reveladas pero que serán evidentes en las acciones que se ejecutarán en las próximas horas. (Translation) The crisis committee, the president assured, has made strategic decisions that, for security, cannot be disclosed but which will be evident in the actions that will be carried out in the next few hours. **Edit:** El comité de crisis ha tomado decisiones que no pueden ser reveladas pero serán evidentes en las acciones que se ejecutarán. (Translation) The crisis committee has made decisions that cannot be disclosed but will be evident in the actions to be carried out.
Init: 大型で非常に強い台風16号は、10月1日の明け方以降、非常に強い勢力で伊豆諸島にかなり近づく見込みです。 (Translation) Very strong typhoon No.16 with a large scale is expected to closely approach to the Izu Islands with a very strong force after the dawn of October 1. **Edit:** 台風16号は伊豆諸島に近づく見込みです。 (Translation) Typhoon No.16 is expected to approach to the Izu Islands.

Table 16: Cases for NDD-based compression on sentences in Chinese, Spanish and Japanese.

Compressing Cases in Multiple Languages Cases in Table 16 show our algorithm to be pretty well-performed on compression of other languages.

ACL 2023 Responsible NLP Checklist

A For every submission:

□ A1. Did you describe the limitations of your work?
Left blank.

□ A2. Did you discuss any potential risks of your work?
Left blank.

□ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

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

B □ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
Left blank.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

□ 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)?
Left blank.

□ 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?
Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

□ 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.
Left blank.

C □ Did you run computational experiments?

Left blank.

□ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

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

□ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

□ 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?
Left blank.

□ 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.)?
Left blank.

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.?
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)?
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?
Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.