Contextualized Semantic Distance between Highly Overlapped Texts

Letian Peng\textsuperscript{1}, Zuchao Li\textsuperscript{2,∗}, and Hai Zhao\textsuperscript{3,∗}

\textsuperscript{1}Department of Computer Science and Engineering, University of California San Diego
\textsuperscript{2}National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, Wuhan, 430072, P. R. China
\textsuperscript{3}Department of Computer Science and Engineering, Shanghai Jiao Tong University

dlepeng@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 adaptation 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.\textsuperscript{1}

\textsuperscript{∗}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. 2042023kfo133).

\textsuperscript{1}Our code is released at github.com/Stareru/NeighboringDistributionDivergence/

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 ($\Delta$PPL) to describe the semantic distance. Unfortunately, $\Delta$PPL 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 \textit{the} and \textit{heavy} in both sentences since the encoder can see these words and encode with their information. But

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.
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 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) \( W = [w_1, w_2, \ldots, w_n] \), 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.

\[
W_m = [w_1, \ldots, w_{i-1}, [\text{MASK}], w_{i+1}, \ldots, w_n]
\]

\[
Q = \text{PLM}^{\text{MLM}}(W_m), q_i = \text{softmax}(Q_i) \in \mathbb{R}^c
\]

\[
p_i = q_i_{\text{Idx}(w_i)}, \text{PPL} = \frac{1}{n} \sum_{i=0}^{n} -\log(p_i)
\]

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.

\[
R_x = \text{PLM}(W_x), R_y = \text{PLM}(W_y)
\]

\[
S_C(W_x, W_y) = \frac{R_x^{\text{CLS}} \cdot R_y^{\text{CLS}}}{||R_x^{\text{CLS}}|| \times ||R_y^{\text{CLS}}||}
\]

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:

\[
q_i = \text{MLM}(W, i), q'_i = \text{MLM}(W', i)
\]

\[
NDD = \sum_{w \in W_{LCS}} a_w F_{\text{div}}(q_{\text{Idx}^w}(w), q'_{\text{Idx}^w}(w))
\]

In this equation, \( F_{\text{div}}(\cdot) \) symbolizes a divergence function that calculates the divergence between distributions. The functions \( \text{Idx}^w(\cdot) \) and \( \text{Idx}^{w'}(\cdot) \) 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

<table>
<thead>
<tr>
<th>Metric</th>
<th>Syn-Ant</th>
<th>POS</th>
<th>Term</th>
<th>Lemma</th>
<th>Sup.</th>
</tr>
</thead>
<tbody>
<tr>
<td>∆PPL</td>
<td>5.3</td>
<td>2.8</td>
<td>8.7</td>
<td>7.9</td>
<td>11.2</td>
</tr>
<tr>
<td>SC</td>
<td>7.8</td>
<td>8.1</td>
<td>9.1</td>
<td>0.0</td>
<td>20.8</td>
</tr>
<tr>
<td>NDD</td>
<td><strong>19.1</strong></td>
<td><strong>22.7</strong></td>
<td><strong>11.2</strong></td>
<td><strong>17.8</strong></td>
<td><strong>24.0</strong></td>
</tr>
<tr>
<td>NDD + SC</td>
<td>12.5</td>
<td>14.5</td>
<td>10.8</td>
<td>6.8</td>
<td>28.2</td>
</tr>
</tbody>
</table>

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 $> x$.

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$^\text{base}$ 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.

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

The STS experiment results are presented in Table 1. For a fair comparison, Roberta$^\text{base}$ is also applied to calculate SC 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. 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 $L_{\text{max}}$ for deletion. Then we use NDD to score the semantic difference caused by the deletions.

---

1Linguistic features in this paper are gotten by models from SpaCy. https://spacy.io/
Figure 4: The compressing scenario of our NDD-based algorithm.

Table 2: Time complexity of evaluating and compressing methods.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVAL ΔPPL</td>
<td>2n</td>
</tr>
<tr>
<td>EVAL S_c</td>
<td>2</td>
</tr>
<tr>
<td>EVAL NDD</td>
<td>2Len(W_{LCS})</td>
</tr>
<tr>
<td>PPL Deleter</td>
<td>O(n^3)</td>
</tr>
<tr>
<td>NDD</td>
<td>O(n^3)</td>
</tr>
<tr>
<td>NDD w/ syn.</td>
<td>O(n^2)</td>
</tr>
<tr>
<td>Fast NDD</td>
<td>O(n^2)</td>
</tr>
<tr>
<td>Fast NDD w/ syn.</td>
<td>O(n)</td>
</tr>
</tbody>
</table>

\[
W_{ij} = [w_1, \ldots, w_{i-1}, w_{j+1}, \ldots, w_n]
\]

\[
NDD_{ij} = NDD(W, W_{ij})
\]

As in Figure 4, We first filter \(W_{ij}\) with \(NDD_{ij}\) above the threshold \(N_{max}\). 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. 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 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.

Configuration We take cased BERT\textsubscript{base} as the PLM for English and BERT\textsubscript{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.

\[
D_{KL}(q, q') = \sum_{k=1}^{c} q_k' \log \left( \frac{q_k'}{q_k} \right) \sim [0, \infty)
\]

We fix the following hyperparameters during the experiment. \(l_{max}\) is set to 9 when the syntax is used and else 5. \(N_{max}\) 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 \(\Delta C = CR_{pred} - CR_{gold}\), which is better when being closer to 0.

---

\[\text{https://www.jamesclarke.net/research/resources}\]
\[\text{https://github.com/ZikangLi/SOM-NCSCM}\]
\[\text{Can be found in Appendix D}\]
### 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Training Data</th>
<th>ROUGE</th>
<th>CR &amp; ∆C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
<td>R1</td>
<td>R2</td>
</tr>
<tr>
<td>Unedited</td>
<td>58.2</td>
<td>63.8</td>
<td>53.4</td>
</tr>
<tr>
<td>SUPERVISED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSTM (Filippova et al., 2015)</td>
<td>80.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LSTM-Dep† (Filippova et al., 2015)</td>
<td>81.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Evaluator-SLM‡ (Zhao et al., 2018)</td>
<td>85.1</td>
<td>78.1</td>
<td>69.9</td>
</tr>
<tr>
<td>Tagger+BERT† (Kamigaito and Okumura, 2020)</td>
<td>85.0</td>
<td>79.3</td>
<td>71.4</td>
</tr>
<tr>
<td>SLAHAN‡ (Kamigaito and Okumura, 2020)</td>
<td>85.5</td>
<td>79.3</td>
<td>71.4</td>
</tr>
<tr>
<td>UNSUPERVISED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drop Head</td>
<td>31.7</td>
<td>23.6</td>
<td>14.7</td>
</tr>
<tr>
<td>Drop Tail</td>
<td>56.0</td>
<td>58.2</td>
<td>47.4</td>
</tr>
<tr>
<td>Random Drop</td>
<td>42.3</td>
<td>41.7</td>
<td>13.6</td>
</tr>
<tr>
<td>PPL Deleter (Niu et al., 2019)</td>
<td>50.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>NDD (Ours)</td>
<td>61.2</td>
<td>60.3</td>
<td>43.2</td>
</tr>
<tr>
<td>NDD+SC‖ (Ours)</td>
<td>62.3</td>
<td>62.6</td>
<td>45.9</td>
</tr>
<tr>
<td>Fast NDD (Ours)</td>
<td>59.7</td>
<td>55.5</td>
<td>40.3</td>
</tr>
<tr>
<td>Fast NDD+SC‖ (Ours)</td>
<td>67.1</td>
<td>62.0</td>
<td>46.6</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Training Data</th>
<th>ROUGE</th>
<th>CR &amp; ∆C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
<td>R1</td>
<td>R2</td>
</tr>
<tr>
<td>SLAHAN‖</td>
<td>✓</td>
<td>✓</td>
<td>57.7</td>
</tr>
<tr>
<td>Fast NDD+SC‖</td>
<td>×</td>
<td>×</td>
<td>76.5</td>
</tr>
</tbody>
</table>

### Baseline
We use the PPL Deleter (Niu et al., 2019) as the main baseline. Deleter uses ∆PPL 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 ∆C 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 ~ 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 & ∆C  
--- & --- & --- & --- & --- & ---  
Unedited & 81.7 & 74.4 & 73.7 & 71.4 & 68.5 & 70.4 & 55.3 & 70.2 & 1.00 (+0.32)  
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)  
PPL Deleter$^1$ & 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$^1$ (Ours) & 76.4 & 68.5 & 59.3 & 52.8 & 47.2 & 69.3 & 49.9 & 69.2 & 0.68 (-0.00)  
NDD (wwm)+SC$^1$ (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$^1$ (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$^1$ (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.

<p>| $N_{max}/L_{max}$ | F &amp; ROUGE &amp; CR &amp; CR &amp; ∆C |</p>
<table>
<thead>
<tr>
<th>---</th>
<th>---</th>
<th>---</th>
<th>---</th>
<th>---</th>
</tr>
</thead>
</table>
| 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 (-0.18)  
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 F1 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.

Compress 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-$k$.

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

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

$^7$https://huggingface.co/hfl/chinese-roberta-wwm-ext

$^8$In analyses, we continue using the KL divergence as the divergence function.
The boy that cake

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

<table>
<thead>
<tr>
<th>Method</th>
<th>$N_{\text{max}}$</th>
<th>Depth-$n$</th>
<th>Subtree-$k$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1 2 3 2 3</td>
<td>≥4 1 2 ≥3 3</td>
</tr>
<tr>
<td>Random</td>
<td>1.0</td>
<td>4</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>4</td>
<td>24</td>
</tr>
<tr>
<td>PPL</td>
<td>1.0</td>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>3</td>
<td>23</td>
</tr>
<tr>
<td>NDD</td>
<td>1.0</td>
<td>1</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>2</td>
<td>23</td>
</tr>
</tbody>
</table>

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

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

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 (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. We apply cased Spanish (SPA) (Joshi et al., 2020) and cased BERT $\text{base}$ (Cañete et al., 2020) as PLMs. For comparison, we implement a PPL-based algorithm that uses $\Delta\text{PPL}$ to detect predicates.

Our results are presented in Table 8. The generally poor performance shows that $\Delta\text{PPL}$ 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 ~ 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.

### Related Works

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

---

$\text{STOP}$

9Instances with length $\leq 50$, number of predicates $> 0$.  
10$a$ for ENG-ID, $th$ for ENG-OOD and $el$ for SPA  
11https://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 ∆PPL for compression, which is not so capable as NDD in semantics preservation.

Syntax and semantic analyses (Dozat and Manning, 2017; Li et al., 2020) 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


A Dataset Statistics

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Google</td>
<td>1000</td>
<td>27.71</td>
<td>0.44</td>
</tr>
<tr>
<td>BNC</td>
<td>595</td>
<td>31.15</td>
<td>0.71</td>
</tr>
<tr>
<td>Colloquial SC</td>
<td>150</td>
<td>8.13</td>
<td>0.68</td>
</tr>
<tr>
<td>STS-B</td>
<td>2758</td>
<td>9.81</td>
<td>-</td>
</tr>
<tr>
<td>PTB</td>
<td>2416</td>
<td>23.46</td>
<td>-</td>
</tr>
<tr>
<td>Conll09-ENG-ID</td>
<td>2399</td>
<td>24.04</td>
<td>-</td>
</tr>
<tr>
<td>Conll09-ENG-OOD</td>
<td>425</td>
<td>16.96</td>
<td>-</td>
</tr>
<tr>
<td>Conll09-SPA</td>
<td>1725</td>
<td>29.35</td>
<td>-</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Sentence</th>
<th>PPL</th>
<th>NDD</th>
<th>s(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I am walking in the cold rain.</td>
<td>5.99</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>I am walking in the cool rain.</td>
<td>10.10</td>
<td>0.81</td>
<td>0.995</td>
</tr>
<tr>
<td>I am walking in the freezing rain.</td>
<td>5.63</td>
<td>0.97</td>
<td>0.997</td>
</tr>
<tr>
<td>I am walking in the heavy rain.</td>
<td>5.30</td>
<td>1.82</td>
<td>0.994</td>
</tr>
<tr>
<td>I am walking in the hot rain.</td>
<td>14.77</td>
<td>3.17</td>
<td>0.995</td>
</tr>
<tr>
<td>I am walking in the cold snow.</td>
<td>5.37</td>
<td>2.46</td>
<td>0.996</td>
</tr>
<tr>
<td>I am walking in the cold night.</td>
<td>6.18</td>
<td>3.52</td>
<td>0.991</td>
</tr>
<tr>
<td>I am walking in the cool sunshine.</td>
<td>8.59</td>
<td>4.73</td>
<td>0.994</td>
</tr>
<tr>
<td>I am running in the cold rain.</td>
<td>11.86</td>
<td>0.66</td>
<td>0.990</td>
</tr>
<tr>
<td>I am wandering in the cold rain.</td>
<td>16.89</td>
<td>0.89</td>
<td>0.982</td>
</tr>
<tr>
<td>I am swimming in the cold rain.</td>
<td>14.84</td>
<td>3.29</td>
<td>0.986</td>
</tr>
<tr>
<td>I was walking in the cold rain.</td>
<td>10.32</td>
<td>4.72</td>
<td>0.980</td>
</tr>
<tr>
<td>He is walking in the cold rain.</td>
<td>105.55</td>
<td>13.04</td>
<td>0.991</td>
</tr>
<tr>
<td>He is walking in the cold rain.</td>
<td>13.95</td>
<td>7.22</td>
<td>0.980</td>
</tr>
</tbody>
</table>

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 freezer 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
for distance.

\[ a_k = \mu \min(|k-i|,|k-j|) \]
\[ a'_k = a_k + a_{n' - k} \cdot \mu^{n'} \]
\[ n' = n - (j - i + 1) \]

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.

\[ b_w = \nu^{Idx(w)} \]
\[ NDD = \sum_{w \in W_{LCS}} a'_w b'_w F_{div}(q_{Idx^d(w)}, q'_{Idx^d(w)}) \]

In experiments, we fix \( \mu \) to 0.9 and adjust \( \nu \) to adapt to the compression rate.

Table 12: Extra comparison including \( S_C \) and BERTScore. \(*\): use \( kl(d||d') \) instead of \( kl(d'||d) \)

<table>
<thead>
<tr>
<th>Method</th>
<th>F</th>
<th>ROUGE</th>
<th>CR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
<td>R1</td>
<td>R2</td>
</tr>
<tr>
<td>PPL Deleter</td>
<td>50.9</td>
<td>51.3</td>
<td>36.7</td>
</tr>
<tr>
<td>PPL Deleter+SC (^*)</td>
<td>53.1</td>
<td>54.7</td>
<td>40.3</td>
</tr>
<tr>
<td>( S_C )</td>
<td>46.5</td>
<td>45.0</td>
<td>16.9</td>
</tr>
<tr>
<td>( S_C+SC )</td>
<td>49.5</td>
<td>50.5</td>
<td>23.0</td>
</tr>
<tr>
<td>BERTScore</td>
<td>44.5</td>
<td>46.5</td>
<td>15.8</td>
</tr>
<tr>
<td>BERTScore+SC (^*)</td>
<td>47.2</td>
<td>49.5</td>
<td>18.8</td>
</tr>
<tr>
<td>( NDD^* )</td>
<td>60.1</td>
<td>59.8</td>
<td>41.7</td>
</tr>
<tr>
<td>( NDD+SC )</td>
<td>60.8</td>
<td>61.9</td>
<td>44.5</td>
</tr>
<tr>
<td>( NDD+SC )</td>
<td>61.2</td>
<td>60.3</td>
<td>43.2</td>
</tr>
<tr>
<td>( NDD+SC ) (^*)</td>
<td>62.3</td>
<td>\textbf{62.6}</td>
<td>45.9</td>
</tr>
<tr>
<td>Fast NDD</td>
<td>59.7</td>
<td>55.5</td>
<td>40.3</td>
</tr>
<tr>
<td>Fast NDD+SC (^*)</td>
<td>\textbf{67.1}</td>
<td>62.0</td>
<td>\textbf{46.6}</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Mask</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>[业务名词]</td>
<td>某业务</td>
</tr>
<tr>
<td>[电话号码]</td>
<td>电话</td>
</tr>
<tr>
<td>[编号]</td>
<td>1</td>
</tr>
<tr>
<td>[名词]</td>
<td>这个</td>
</tr>
<tr>
<td>[地址]</td>
<td>某地</td>
</tr>
</tbody>
</table>

D Mask to Expression

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPL Deleter</td>
<td>3.69</td>
<td>2.56</td>
</tr>
<tr>
<td>PPL Deleter+SC†</td>
<td>3.93</td>
<td>2.88</td>
</tr>
<tr>
<td>SC</td>
<td>1.87</td>
<td>1.51</td>
</tr>
<tr>
<td>SC+SC†</td>
<td>2.35</td>
<td>2.08</td>
</tr>
<tr>
<td>BERTScore</td>
<td>1.96</td>
<td>1.83</td>
</tr>
<tr>
<td>BERTScore+SC†</td>
<td>2.38</td>
<td>2.17</td>
</tr>
<tr>
<td>NDD</td>
<td>3.87</td>
<td>3.46</td>
</tr>
<tr>
<td>NDD+SC†</td>
<td>4.08</td>
<td>3.62</td>
</tr>
<tr>
<td>Fast NDD</td>
<td>3.73</td>
<td>3.18</td>
</tr>
<tr>
<td>Fast NDD+SC†</td>
<td>4.01</td>
<td>3.43</td>
</tr>
</tbody>
</table>

• 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**

<table>
<thead>
<tr>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit: The speed limit will be 70 mph despite opposition from organizations.</td>
</tr>
<tr>
<td>Gold: The speed limit on highways in Illinois will be raised to 70 mph next year.</td>
</tr>
<tr>
<td>F1 Score = 51.9 (↓ 8.5)  ROUGE = 53.8</td>
</tr>
</tbody>
</table>

**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.

**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.

<table>
<thead>
<tr>
<th>Iter: 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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final: A mill has been inaugurated to boost aquaculture sector.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 15: Cases for output in iterations of the NDD-based compression. <strong>Bold: Kept components</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Outputs from Compression Iterations</strong>  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.</td>
</tr>
</tbody>
</table>
(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.

(Translation) During the price adjustment, Saudi scales down the price of crude oil, our country releases the reserved crude oil.

(Translation) The crisis committee has made decisions that cannot be disclosed but will be evident in the actions to be carried out.

(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.

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.

10930
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.