Abstract

Over the last few years, Masked Language Modeling (MLM) pre-training has resulted in remarkable advancements in many Natural Language Understanding (NLU) tasks, which has sparked an interest in researching alternatives and extensions to the MLM objective. In this paper, we tackle the absence of explicit semantic grounding in MLM and propose Descriptive Masked Language Modeling (DMLM), a knowledge-enhanced reading comprehension objective, where the model is required to predict the most likely word in a context, being provided with the word’s definition. For instance, given the sentence “I was going to the _”, if we provided as definition “financial institution”, the model would have to predict the word “bank”; if, instead, we provided “sandy seashore”, the model should predict “beach”. Our evaluation highlights the effectiveness of DMLM in comparison with standard MLM, showing improvements on a number of well-established NLU benchmarks, as well as other semantics-focused tasks, e.g., Semantic Role Labeling. Furthermore, we also demonstrate how it is possible to take full advantage of DMLM to embed explicit semantics in downstream tasks, explore several properties of DMLM-based contextual representations and suggest a number of future directions to investigate.

1 Introduction

Language Modeling is at the core of transfer learning approaches that have recently revolutionized the Natural Language Processing field. Among these, the Masked Language Modeling (MLM) formulation introduced by Devlin et al. (2019) has been used to train Large Language Models that obtained astounding performances in many Natural Language Understanding (NLU) tasks. This proved empirically that training a model to predict a word based on the context in which it appears (i.e., the cloze task; Taylor, 1953) enables the emergence of rich word representations with transferable value (Ruder et al., 2019).

Given the importance of MLM, several improvements over its standard formulation have been proposed. In particular, a large body of research has investigated alternative self-supervised objectives to take advantage of the ever-growing availability of raw text. For instance, Dong et al. (2019) unified multiple language modeling objectives into a single architecture, Joshi et al. (2020) masked entire spans instead of single tokens, and Clark et al. (2020) exploited the entire input instead of only optimizing on the masked words. At the same time, another research direction has explored utilizing, alongside MLM, the wealth of information contained in structured Knowledge Bases (KBs) to enhance models’ representations. For example, Peters et al. (2019) and Liu et al. (2020b) tried leveraging KB entities in order to provide additional input context, while Levine et al. (2020) and Yamada et al. (2020) tasked the models with explicitly predicting KB-grounded embeddings of concepts and named entities in place of words, as an integration to the MLM framework.

Our work stands in the middle of the aforementioned directions. Indeed, we have designed a semantic-enhanced objective that is able to semantically ground word representations and provide complementary information to the cloze task without ever leaving the MLM framework. Specifically, we put forward Descriptive Masked Language Modeling (DMLM), a pre-training objective that requires the model to perform reading comprehension over textual definitions: given an input sentence, the model is tasked with predicting a masked word in context while being provided with a natural language definition, drawn from a predefined sense inventory, that describes its meaning.

At the same time, given that this auxiliary task...
I went [MASK] the [DEF]. Sandy [MASK]
beach to seashore
I went [MASK] the [DEF]. Financial [MASK]
bank to institution

Figure 1: Two examples of DMLM, where a model has to predict a described masked word while simultaneously performing standard MLM.

is orthogonal to MLM, in that it uses descriptions, rather than the surrounding context, to help models guess the correct word, and in order to improve optimization efficiency and contextualization capabilities, we utilize the standard MLM objective over the entire input sequence, definition included (Figure 1).

While our primary focus is to overcome the absence of explicit semantic grounding in MLM, thus producing semantically rich representations that can later be used in downstream applications, we show that, as a by-product of our objective, DMLM-trained models can, i) leverage DMLM’s objective in downstream tasks, and, ii) exhibit grounding towards the sense inventories involved in the pre-training. Furthermore, we demonstrate that Word Sense Disambiguation systems (Bevilacqua et al., 2021; Navigli, 2009) can be employed effectively to produce large-scale sense-tagged corpora, dropping the need for manually disambiguating millions of words in order to train DMLM systems.

To summarize, our contributions are manifold:

• We introduce Descriptive Masked Language Modeling (DMLM), a novel knowledge-enhanced reading comprehension objective;
• We extensively evaluate architectures trained with DMLM over GLUE, i.e., a set of NLU tasks, as well as on a semantics-focused downstream task, i.e., Semantic Role Labeling, showing that DMLM-trained models consistently outperform their MLM counterparts;
• We show that DMLM-trained models can take advantage of word definitions in downstream tasks, even if these latter were not seen during pre-training, opening several possibilities for semantically enriching contexts;
• Through a spatial analysis, we show that the word representations produced by a DMLM-trained encoder partially exhibit grounding towards the KBs employed during training.

We release code as well as all sense-tagged corpora used for training at https://github.com/SapienzaNLP/dmlm.

2 Related Work

Since the advent of BERT (Devlin et al., 2019), Masked Language Modeling (MLM) has been used widely to pre-train language models in a self-supervised fashion. As opposed to standard Language Modeling, which requires predicting the next word in a sequence given the preceding words, MLM consists in predicting masked words given the remaining context.

MLM Revisions & Extensions Over the past few years, several extensions to MLM have been proposed (Qiu et al., 2020). To name a few, Dong et al. (2019, UniLM) introduced an ensemble of pre-training objectives to unify masked, causal and sequence-to-sequence language modeling. Joshi et al. (2020, SpanBERT) proposed an extension that masked and predicted entire spans, forcing the model to predict them solely based on the context, which is arguably harder than predicting single masked words. More recently, Clark et al. (2020, ELECTRA) improved MLM’s efficiency by optimizing over all the tokens of the sequence, using a generator to perturb the input sentence and a discriminator that needs to discern between original and modified tokens. Finally, several works have cast MLM to the sequence-to-sequence setting, applying different masking techniques to both the input and the output sequences (Song et al., 2019; Lewis et al., 2020; Raffel et al., 2020).

Knowledge-Enhanced Pre-training Although the self-supervised MLM objective has proven to model both syntactic and semantic information (Rogers et al., 2020), its formulation provides no explicit ties to the real world (Peters et al., 2019; Zhang et al., 2019), a limitation that several works have tried to overcome through the injection of information coming from Knowledge Bases (KBs). Some works proposed to extend the output vocabulary of MLM with either entities in Wikipedia (Yamada et al., 2020, LUKE) or supersenses found in WordNet (Levine et al., 2020, SenseBERT), while Peters et al. (2019, KnowBERT) leveraged entity embeddings computed from either Wikipedia or
WordNet (Miller et al., 1990) to re-contextualize the output representations of the underlying model.

Another research direction focused on enhancing the input sequence provided to the underlying model. For instance, Liu et al. (2020b, K-BERT) added a KB module to retrieve relevant entities and relations, injecting them into the sentence, while Wang et al. (2021, KEPLER) used an encoder model to jointly learn entity embeddings through their corresponding descriptions and perform the standard MLM objective.

Finally, and much closer to our work, Chen et al. (2022, DictBERT) used entries of the Cambridge dictionary (words and their definitions) to produce latent vectors that enhanced models’ hidden representations, while Yu et al. (2022, Dict-BERT) helped models better contextualize rare words by appending their definitions (taken from Wiktionary) to the input sequence, though they prevented any other word in the input from taking advantage of the definitions provided.

In stark contrast to previous efforts, we put forward a novel auxiliary task to MLM, where the model is required to predict a masked word based on both the context it appears in and, most importantly, its definition, which we extract from dictionary-like KBs; at the same time, the model is also trained to perform MLM over the entire input sequence, including the definition. Our method has two inherent advantages: first, that it is not restrained by a fixed vocabulary, and, second, that it supports semantic enrichment via the injection of definitions, including those not seen during pre-training, in downstream tasks. Moreover, as an additional benefit of leveraging DMLM alongside MLM, we show that DMLM embeds properties related to the Knowledge Bases employed during training.

3 Descriptive MLM

As a first step, let us formally define the task of Masked Language Modeling (MLM): given a sequence of \( n \) words \( w_1, \ldots, w_n \), we randomly replace a certain percentage of the words by means of a special, uninformative, \([\text{MASK}]\) token, and ask the model to predict the corresponding masked words. For example, given the sentence “I went to the beach.”, if we picked the word beach randomly, the model would see “I went to the [\text{MASK}].” as input, and would be asked to predict beach from its corresponding \([\text{MASK}]\) token.

DMLM builds on top of MLM in that, before applying MLM, i) it randomly selects a content word \( w_i \) from the input sentence, for which we know its textual definition \( d^{w_i} \), ii) it replaces \( w_i \) with another special token, i.e., \([\text{DEF}]\), and, iii) it appends \( d^{w_i} \) to our input sequence after a special \([\text{DEFINE}]\) token. Going back to our example, we would obtain “I went to the \([\text{DEF}]\). [\text{DEFINE}] Sandy seashore”. After this initial step, we apply the standard MLM perturbation avoiding replacing either of the two special tokens that we added, but leaving \( d^{w_i} \) as a possible target for perturbation, e.g., “I went [\text{MASK}] the [\text{DEF}].. [\text{DEFINE}] Sandy [\text{MASK}]” (see Figure 1). Following this procedure, depending on which tokens are masked, the model has to leverage, potentially simultaneously, both the input sequence and the content word’s definition in order to restore the masked words. Thus, utilizing DMLM implies that:

- Every word in the input sequence is used to re-contextualize both the \([\text{MASK}]\) words and the special \([\text{DEF}]\) word;
- The model, especially in ambiguous contexts, must exploit the definition to predict the \([\text{DEF}]\) token;
- The definition is perturbed as well, so that it not only contributes to the prediction of the masked content word, but also requires the model to use all unmasked tokens, even those in the input sequence, to reconstruct that definition.

DMLM can be used to pre-train both encoder-only and encoder-decoder architectures. For encoder-only architectures, when a content word is split into subwords, we replicate the \([\text{DEF}]\) token for each subword so that the model is able to reconstruct the full word at prediction time. For encoder-decoder architectures, DMLM’s formulation can easily be applied with very small adjustments. Due to space and computational constraints, we discuss these adjustments in more detail in Appendix A.

4 DMLM Pre-training

4.1 Sense-tagged Dataset

Training a system using the DMLM objective requires a large corpus whose content words are

\(^1\)Words, or multi-words, with open-class Part-of-Speech tags, i.e., nouns, verbs, adverbs, and adjectives.

\(^2\)We also support multi-word expressions.
Table 1: Statistics about the produced sense-tagged corpora. Tokens: total number of tokens in the corpus; it is the same underlying corpus for all inventories (WikiText-103). Instances: number of content words that have an associated sense and, therefore, definition. Token Vocab.: number of distinct lemmas that have an associated definition. Definitions: number of distinct definitions associated with the instances.

<table>
<thead>
<tr>
<th>Inventory</th>
<th>Tokens</th>
<th>Instances</th>
<th>Token Vocab.</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>WordNet</td>
<td>98 M</td>
<td>44 M</td>
<td>224 K</td>
<td>89 K</td>
</tr>
<tr>
<td>ODE</td>
<td>98 M</td>
<td>72 M</td>
<td>84 K</td>
<td>79 K</td>
</tr>
<tr>
<td>Wiktionary</td>
<td>98 M</td>
<td>69 M</td>
<td>183 K</td>
<td>91 K</td>
</tr>
</tbody>
</table>

Definition inventory breakdown of the tagged corpus.

paired with suitable definitions. To create such a corpus, we leveraged Word Sense Disambiguation (WSD), i.e., the task of identifying the most appropriate meaning of a word in a given context from a predefined sense inventory (Bevilacqua et al., 2021). Specifically, we employed ESCHER (Barba et al., 2021), a high-performance WSD system, and disambiguated the whole WikiText-103 corpus (Merity et al., 2017). In this work, we used the following inventories which, by design, come with a definition for each of their senses:

- **WordNet** (Miller et al., 1990), the most commonly used English sense inventory for WSD. Following the literature, we use WordNet 3.0;

- **ODE**, the Oxford Dictionary of English inventory as provided by Chang et al. (2018);

- **Wiktionary**, a sense inventory containing senses from the English Wiktionary project. We used a polished dump from November 2021 using the same preprocessing pipeline as in Bevilacqua et al. (2020).

Following the reference paper, we trained ESCHER jointly on all three inventories, obtaining results that were comparable to the original model.4

Using the resulting model, we tagged the entire WikiText-103 corpus, at sentence level, three times, once for each inventory, which we posit acts as a regularization factor for the DMLM objective. Indeed, the model might encounter the same content word in the exact same sentence but with different definitions representing the same meaning,5 thus reducing overfitting on the definitions themselves.

In the end, the disambiguated corpus contained around 3.8M sentences with a total of approximately 381K unique definitions coming from the three different inventories. Table 1 provides a per-inventory breakdown of the tagged corpus.

4.2 Model Architectures

As our underlying model, we followed the BERT architecture (Devlin et al., 2019), with the exception of the number of layers and attention heads, which we restricted to 6 and 8, respectively; furthermore, we experimented with three different hidden sizes, i.e., 256, 512 and 768, resulting in three models with around 20M, 43M and 66M parameters.

While, due to computational constraints, we had to train relatively small architectures compared to current trends in NLP, we performed a small-scale study of the impact of network size to give a rough idea of how DMLM could fare on larger architectures. We hope that the encouraging results we report in this work will foster research in this direction, especially in investigating the effectiveness of DMLM on larger networks.

4.2.1 Experiment runtimes

Pre-training our architectures, with the setup described in Section 4.3, required around 5 days for the two 43M models, 2.5 days for the 20M model and 8.5 days for the 66M model.

As for the downstream tasks, we used a NVIDIA RTX 3090 for fine-tuning. On GLUE and WiC, our architectures required around 6h per model, 9h for DistilBERT, 12h for BERTbase and around 24h for BERTlarge. For SRL, at training time, our models each took around 1h40m without and 2h15m with definitions appended, while larger models took up to 3h30m without and 4h30m with definitions. As far as the inference speed of DMLM models for SRL was concerned, when appending predicate definitions to improve the model accuracy (Figure 2b), evaluating over the entire CoNLL-2009 test set instances took 1m37s, around 36% slower than when not using the definitions (1m11s).

4.3 Pre-training procedure

We trained our networks with an overall batch size of 256 sentences on 4 NVIDIA 40GB A100 GPUs in BFLOAT16 (Dean et al., 2012) half-precision

---

4https://en.wiktionary.org/

5Factoring in that different sense inventories might exhibit different sense granularities.
format. We used Rectified Adam (Liu et al., 2020a) as optimizer with a learning rate of $10^{-5}$. We limited the number of maximum training steps to 1,000,000 and evaluated overall performance on a held-out validation dataset every 30,000 steps, which we also used for model selection.

During training, while all sentences were subject to MLM, DMLM was only applied with probability 0.5, so as to increase the model’s robustness to the absence of definitions, which is the most common setting in fine-tuning scenarios. Finally, when applying random masking in both objectives, we followed Devlin et al. (2019) and masked 15% of the subwords in input, replacing them 80% of the time with the [MASK] token, 10% of the time with a random word and 10% of the time keeping the unmodified original subword.

5 Experimental Evaluation

In order to assess the quality of the trained models, we evaluated them on a number of different, both semantic and non-semantic, downstream tasks. Each of the following subsections, aside from Section 5.1 which describes the comparison systems, contains the setup and results for each experiment we performed.

5.1 Comparison Systems

To assess the improvements brought by DMLM, we trained our encoder both with and without our auxiliary objective. We used a 43M parameter model to compare MLM and DMLM directly, while we also trained two additional 20M and 66M parameter models to assess the impact of architecture scaling on DMLM (see Section 4.2). While we used the WikiText-103 corpus for both objectives, the DMLM models had access to the definitions of the disambiguated words, thus increasing the total number of tokens processed at training time. To account for this, when training such models, we removed as many sentences as needed to reach the same number of tokens the MLM model was trained on, while we maintained the same mean and variance for the input sequences length.5

Furthermore, since we were not able to train two additional 20M and 66M MLM-only models for comparison, we took both DistilBERT (Sanh et al., 2019) and TinyBERT (Jiao et al., 2020) as direct competitors of our 66M model, as they have the same number of parameters, while we compared BERT$_{small}$ (Bhargava et al., 2021, 29M parameters) against DMLM$_{20M}$. It is important to note that, despite having similar parameter counts, these models are a product of distillation, which leverages the training of a much larger model and generally produces models that perform better than ones trained from scratch (Turc et al., 2019). Nevertheless, we will show that our models still outperform them in almost any tested setting.

Finally, as additional reference baselines, we computed and report here results achieved by both BERT$_{base}$ and BERT$_{large}$ (Devlin et al., 2019, 110M and 335M parameters respectively). We do not include the knowledge-enhanced models described in Section 2, since some of them start from a pre-trained model, and some are not comparable in size. Nevertheless, we report their performances on the GLUE benchmark, where available, in the Appendix (Table 6).

5.2 GLUE

The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018) is a collection of diverse Natural Language Understanding tasks, and the de facto standard for the evaluation of pre-trained language models. A detailed breakdown of the tasks is reported in Appendix D.

Setup We perform our experiments using jiant (Phang et al., 2020), the official GLUE toolkit, training with the default hyperparameters. For each task, we fine-tune a copy of the model; moreover, since we are using the GLUE validation dataset to compare different systems, we do not perform any ensembling or parameter tuning, as is commonly done for GLUE submissions (Clark et al., 2020). Following the literature, we do not report results on WNLI as it is difficult to beat even the majority classifier using a standard fine-tuning-as-classifier approach (Devlin et al., 2019; Clark et al., 2020). For SST-2, QQP, MNLI and QNLI we train for 3 epochs and report the results on a single seed. For CoLA, RTE, MRPC and STS-B, which are quite a lot smaller in size, we train for 5 epochs, perform 5 repeated runs with different seeds and report the median value of each task-specific metric.

Results There are a number of considerations to be drawn from the results in Table 2, when comparing DMLM against MLM, and when assessing the impact of network scaling.

---

5For our sense-tagged WikiText-103 corpus, we remove around 17.3% of all sentences.
### 5.3 Semantic Role Labeling

Semantic Role Labeling (SRL) – the task of understanding “who did what to whom, where, when and how?” – is regarded as an inherently semantic task requiring comprehension of the input sentence (Gildea and Jurafsky, 2000). SRL is usually split into four sub-tasks: i) Predicate Identification, where the model sees the input sentence and has to identify the main predicates; ii) Predicate Disambiguation, where the model has to choose the correct meaning for each of the identified predicates among its possible senses; iii) Argument Identification, where the model has to identify which words represent the arguments of the given predicate; iv) Argument Classification, where the model has to classify the identified arguments for the given predicate.

#### Setup
Following Conia andNavigli (2020), Blloshmi et al. (2021) and Shi and Lin (2019), we feed our model with the identified predicate, hence skipping the first step of the SRL pipeline, and perform the remaining three steps simultaneously as in a standard token classification setting.

---

### Table 2: Results on GLUE. Per-task metrics are reported in Appendix D: MRPC and QQP both display F1 / accuracy, while STS-B shows Pearson / Spearman Rank. Underlined task names represent those with repeated runs. Numbers between 0 and 1 are multiplied by 100 to ease readability.

<table>
<thead>
<tr>
<th>Our work</th>
<th>#Params</th>
<th>CoLA</th>
<th>SST-2</th>
<th>MRPC</th>
<th>STS-B</th>
<th>QQP</th>
<th>MNLI$_{mm}$</th>
<th>MNLI$_{sm}$</th>
<th>QNLI</th>
<th>RTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT$_{sm}$</td>
<td>110M</td>
<td>58.57</td>
<td>92.43</td>
<td>88.89</td>
<td>84.31</td>
<td>87.29</td>
<td>90.57</td>
<td>83.42</td>
<td>84.08</td>
<td>90.92</td>
</tr>
<tr>
<td>BERT$_{base}$</td>
<td>335M</td>
<td>62.58</td>
<td>93.46</td>
<td>90.36</td>
<td>86.03</td>
<td>87.96</td>
<td>91.04</td>
<td>85.80</td>
<td>85.84</td>
<td>92.28</td>
</tr>
<tr>
<td>DistilBERT</td>
<td>29M</td>
<td>70.78</td>
<td>87.39</td>
<td>85.03</td>
<td>76.96</td>
<td>84.27</td>
<td>87.85</td>
<td>76.69</td>
<td>36.76</td>
<td>86.49</td>
</tr>
<tr>
<td>TinyBERT</td>
<td>66M</td>
<td>47.45</td>
<td>90.48</td>
<td>87.32</td>
<td>82.35</td>
<td>84.54</td>
<td>89.14</td>
<td>80.47</td>
<td>81.66</td>
<td>87.24</td>
</tr>
<tr>
<td>TinyBERT</td>
<td>66M</td>
<td>44.28</td>
<td>91.51</td>
<td>90.39</td>
<td>86.76</td>
<td>89.30</td>
<td>89.18</td>
<td>83.78</td>
<td>70.97</td>
<td>90.39</td>
</tr>
<tr>
<td>DMLM$_{43M}$</td>
<td>43M</td>
<td>27.02</td>
<td>89.22</td>
<td>79.53</td>
<td>70.10</td>
<td>81.27</td>
<td>84.33</td>
<td>72.56</td>
<td>74.08</td>
<td>76.61</td>
</tr>
<tr>
<td>DMLM$_{66M}$</td>
<td>66M</td>
<td>26.61</td>
<td>90.57</td>
<td>86.35</td>
<td>82.31</td>
<td>86.15</td>
<td>86.12</td>
<td>80.29</td>
<td>79.51</td>
<td>85.16</td>
</tr>
<tr>
<td>DMLM$_{20M}$</td>
<td>20M</td>
<td>27.07</td>
<td>87.54</td>
<td>83.61</td>
<td>74.43</td>
<td>83.94</td>
<td>87.14</td>
<td>78.12</td>
<td>75.12</td>
<td>83.37</td>
</tr>
<tr>
<td>DMLM$_{66M}$</td>
<td>66M</td>
<td>53.13</td>
<td>91.92</td>
<td>90.51</td>
<td>86.89</td>
<td>89.73</td>
<td>89.64</td>
<td>84.65</td>
<td>83.92</td>
<td>89.48</td>
</tr>
</tbody>
</table>

**MLM$_{43M}$ vs DMLM$_{43M}$** Starting from the Natural Language Inference (NLI) tasks, we can see an increase of up to 7.7, 5.4 and 8.5 in F1 score for MNLI$_{mm}$, MNLI$_{sm}$ and QNLI, respectively, confirming the robustness of the representations produced with our objective, even on tasks that are not focused primarily on semantics. As expected, DMLM also consistently outperforms MLM in more semantically-focused tasks. Indeed, on MRPC, STS-B and QQP, all tasks where models are required to measure the semantic equivalence between two sentences, the performance gap remains considerably large, with an increase of up to 5.8 and 16.8 F1 points on QQP and MRPC, respectively; most notably, on STS-B we report an improvement of around 45 F1 points, doubling the score of MLM$_{43M}$, and also surpassing DistilBERT, despite the difference in size. Finally, on RTE, where models have to predict if a premise entails the corresponding hypothesis, DMLM$_{43M}$ outperforms both MLM$_{43M}$ and DistilBERT.

These results suggest that, when trained in comparable settings and at least for the model size we consider, including the DMLM objective in the pre-training results in representations that outperform MLM-only pre-training in many downstream tasks, especially semantics-focused ones.

**DMLM scaling** On the one hand, despite the difference in size, we observe similar performances between our 20M model and BERT$_{small}$ (50% larger). Interestingly, the only task where our model strongly outperforms BERT$_{small}$ is MNLI$_{sm}$, with a 40 points difference, which seems to be related to a lack of generalization in MNLI by BERT$_{small}$.

On the other hand, when comparing our 66M model against its competitors, we observe that DMLM$_{66M}$ consistently outperforms both DistilBERT and TinyBERT on every task except QNLI, with the largest gaps in RTE (8 and 4 points) and CoLA (6 and 9 points), where DMLM$_{66M}$ appears to be large enough for the model to form meaningful grammatical latent structures. Furthermore, we observe that TinyBERT exhibits a behavior similar to BERT$_{small}$ in MNLI, with the MNLI$_{sm}$ lagging behind MNLI$_{mm}$ by 13 points. Moreover, we point out that the performance improvement between the 43M and 66M models is, on average, bigger than the improvement between the 20M and the 43M models, with the largest difference in MNLI$_{sm}$ (75.12$_{20M}$ → 79.51$_{43M}$ → 83.92$_{66M}$), justifying future research efforts in scaling up network sizes.
Specifically, the model receives in input the sentence with two special tokens delimiting the predicate, i.e., [PR] and [/PR] (Figure 2a). Then, for predicate disambiguation, we pass the vector representation corresponding to the first subword of the predicate through a classification layer. Similarly, for argument classification, we pass the remaining vectors through another classification layer, which outputs a distribution over all possible arguments, including the NULL one, and take the arguments tied to the predicted predicate with the highest probability.

We fine-tune models using RAdam (Liu et al., 2020a) with a learning rate of $5 \times 10^{-5}$ and a batch size of 16. We use the datasets provided in CoNLL-2009 (Hajič et al., 2009) and CoNLL-2012 (Pradhan et al., 2012) for training and evaluation. For CoNLL-2009, we use the official scorer\footnote{https://ufal.mff.cuni.cz/conll2009-st/scorer.html} released alongside the dataset, while for CoNLL-2012 we use the scorer that was released for the span-based SRL Shared Task of CoNLL-2005.\footnote{https://www.cs.upc.edu/~sriconll/soft.html}

### Results

In Table 3 we report the results obtained on the three test sets. First, we observe that our baselines are quite effective, as their scores are in the same ballpark as state-of-the-art systems using the same underlying transformer, i.e., BERT\text{large} for Conia and Navigli (2020); Shi and Lin (2019).

Regarding our models, DMLM\text{43M} consistently outperforms MLM\text{43M} by a large margin: this difference stays between 1.2 and 1.9 F1 points, showing that our pre-training objective helps modeling the semantic content of the input sentence. Moreover, DMLM\text{66M} comfortably beats both DistilBERT and TinyBERT, and almost achieves scores around BERT\text{base}, despite being around half its size.

### Leveraging the DMLM objective

While these results are interesting in their own right, the main reason we include SRL in our evaluation is that we can take advantage of DMLM’s pre-training objective. Specifically, since each predicate sense has a corresponding definition in PropBank (Palmer et al., 2005), we investigate how performances change when, mimicking DMLM, we replace the target predicate with the [DEF] token and inject the definition associated with the predicate as disambiguated by the model.\footnote{The prediction in this setting has two phases: first, the model disambiguates the target predicate, then we inject the definition associated with the disambiguated predicate to the input sentence, replace the target predicate with [DEF] and perform argument identification and classification.}

---

**Table 3: Results on Semantic Role Labeling.** Numbers represent F1 scores as computed by the official evaluation scripts for both years. C09 and C12 stand for CoNLL-2009 and CoNLL-2012 respectively; C09\text{OOD} represents the Out-Of-Domain portion of the CoNLL-2009 test set.
in Figure 2b, given the sentence “Mary sold her book to John”, the model would see as input “Mary [PR] [DEF] [/PR] her book to John [DEFINE] Exchange for money” (this formulation applies to systems with the DEF superscript in Table 3).

With this setup, we observe overall improvements, in comparison to MLM, between 2.5 and 4.2 F1 points; further, our 43M model outperforms DistilBERT, reaching scores comparable to BERTbase, which has almost three times the number of parameters. Moreover, to ensure that the performance improvements are not solely due to the additional context provided by the definitions, we append them to MLM-only models (without replacing the target predicate): as a result, we find no meaningful differences, confirming that the DMLM objective gives models the ability to leverage the definition effectively. Additionally, our 66M model manages to compete against state-of-the-art models, despite the wide gap in size (66M vs 330M+), proving the effectiveness of our descriptive pre-training and its potential when dealing with semantic tasks. Furthermore, once again, we observe how increasing the model size results in general improvements, with the largest gaps occurring between the 43M model and the 66M model. To summarize, we find that:

1. DMLM fares better than MLM with semantically-enriched text, and that
2. DMLM-trained models can scale to unseen definitions, as we demonstrated here with PropBank, attesting to the generalization capability of DMLM.

5.4 Word Sense Disambiguation

Since its introduction, BERT’s contextualized representations have been studied thoroughly to assess whether they are semantically coherent with discrete senses coming from external sense inventories, e.g., WordNet (Wiedemann et al., 2019; Scarlini et al., 2020a,b; Loureiro et al., 2021). Indeed, following these works, we perform a very similar analysis of our encoder-only architecture, tackling WSD via 1-NN search: assuming that contextualized representations of a word in a sentence should represent their meaning, we compare the contextual representation of the word we are trying to disambiguate against all the contextual representations of words for which we know the sense, returning the sense associated with the closest one.

Setup We use SemCor (Miller et al., 1993), the standard manually-annotated WSD training corpus with senses coming from WordNet, to generate the reference encodings for words in context and their associated senses (by taking the last hidden state produced by each model), and use the ALL test set introduced by Raganato et al. (2017) for evaluation.10

Results Table 4 reports the results of this analysis. First, regarding the baselines, we observe that the F1 scores correlate with the number of parameters, showing an absolute difference of 2 F1 points between DistilBERT and BERTlarge, and a similar trend with our 20M, 43M and 66M models.

Moving over to our architectures, we find that, even though DMLM does not force the spatial distribution of contextualized words to follow the sense distribution explicitly, this happens to some extent. Indeed, DMLM43M surpasses MLM43M by 3.6 F1 points; interestingly, we perform in the same ballpark as BERTlarge both with DMLM43M and DMLM20M, despite the latter being 4th in size; we posit that the injection of definitions, which the model has always seen during training bound to specific words in one of their meanings, has helped to build representations that more closely relate to WordNet’s senses.

Finally, following the experiment we performed in SRL (Section 5.3), we also create the reference encodings by feeding the model with the [DEF] token in place of the target word, appending the definition associated with the sense as found in SemCor. We report the best results across the board, with DMLM366M achieving, in this setting, 5.4 F1 points more than BERTlarge and a 2.6 F1 improvement over plain DMLM (Table 4), backing our claim that the definitions disambiguate words in context.

In general, regardless of model size, including the definitions results in performance enhancements ranging from 1.6 to 4.2 F1 points.

5.5 Exploring the Spatial Distribution

As a final experiment, we study how close the contextualized representations of words are in comparison to the sense they take upon. To do this, we compute the cosine similarities between different groups of words in the SemCor dataset: i) between words sharing the same sense, to get a grasp of how close sense representations are, ii) between words

---

10We report a detailed explanation of the Word Sense Disambiguation experimental setting in Section C.
Table 4: Left: results on 1-NN WSD, numbers are F1 scores. Right: average similarities (multiplied by 100) computed on different groups of contextualized words.

<table>
<thead>
<tr>
<th>Baselines</th>
<th>WSD ↑</th>
<th>Cosine Similarities</th>
<th>ALL↑</th>
<th>Sense</th>
<th>Lemma-PoS (Δ)</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rescaling</td>
<td>MLM</td>
<td>59.6</td>
<td>56.1</td>
<td>58.4 (-2.3)</td>
<td>28.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DMLM</td>
<td>65.7</td>
<td>83.9</td>
<td>65.9 (+18.0)</td>
<td>31.9</td>
<td></td>
</tr>
<tr>
<td>Scaling</td>
<td>DMLM</td>
<td>62.5</td>
<td>67.3</td>
<td>66.4 (+0.9)</td>
<td>22.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DMLM</td>
<td>64.1</td>
<td>65.4</td>
<td>60.4 (+5.0)</td>
<td>15.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DMLM</td>
<td>64.1</td>
<td>75.8</td>
<td>68.9 (+6.9)</td>
<td>43.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DMLM</td>
<td>68.3</td>
<td>82.4</td>
<td>60.2 (+22.2)</td>
<td>34.3</td>
<td></td>
</tr>
</tbody>
</table>

Results

We report the results of this analysis in Table 4 (Cosine Similarities). Starting with the baselines, we observe that the average cosine similarity between random words decreases as the number of parameters increases, with DistilBERT exhibiting the highest similarity at around 63. Furthermore, the only MLM-trained model where the average similarity of same-sense words is higher than same-lemma and PoS words is BERTlarge.

Regarding DMLM-trained models, on the other hand, we observe two interesting properties: first, the cosine similarity between words sharing the same WordNet sense is higher than that of words that only share the same lemma and POS, a property that we found only in both BERT variants. This supports the hypothesis that DMLM introduces a shift towards the inventories’ senses used during training, even if the DLM objective does not explicitly favor it. Second, the output space shows around the same clustering behavior as BERTlarge, despite the huge gap in pre-training compute and model size; in contrast, distilled models display similar spatial distributions, regardless of their size.

Finally, when replacing the target word with the [DEF] token and including the definition in the input, the model is fully taking advantage of the definition and conveying its meaning in the [DEF] token, as the sense to lemma-PoS difference is the highest among all others by a large margin, while we attribute the higher random similarity to the fact that the underlying token is always [DEF].

6 Conclusions

In this work, we presented an extension of MLM called Descriptive Masked Language Modeling (DMLM), which embeds semantic information via natural language descriptions in the pre-training phase of language models.

We found that, under the tested settings, DMLM consistently outperforms MLM on multiple benchmarks. On the GLUE Benchmark, a set of Natural Language Understanding tasks, we observed improvements on both semantic and non-semantic tasks. Furthermore, using SRL as a proxy, we also demonstrated two important properties of models trained with DMLM: first, that it is possible to leverage DMLM’s pre-training objective to consistently improve performances in downstream tasks and, second, that the model can generalize to definitions that were not seen during pre-training. Finally, we discovered that, even without any explicit signal towards spatial alignment, the output space of a DMLM-trained encoder tends to relate to the Knowledge Bases used to retrieve the definitions. We posit that this might be a very desirable property for better handling ambiguity, e.g., in Machine Translation, where recent works have shed some light on the issue of semantic biases (Campolungo et al., 2022). Additionally, in principle, we could make DMLM-trained systems more suitable for domain-specific tasks. For example, in the medical domain, we could impose precise meanings for word senses based on a healthcare-specific knowledge base and thus reduce the conflation of senses’ representation in the same space. Second, being able to drive spatial representations of words during the training and select a reference knowledge base as a guide might be the very reason why DMLM-trained systems outperform, at least in our experimental setting, MLM-trained systems.

Given the encouraging results obtained while scaling up the network size, and to foster research in this direction, we release our code and the sense-tagged WikiText-103 corpus.

7 Limitations

Model sizes and comparability As we have pointed out in the paper, due to computational and time constraints on the hardware we had at our disposal, we found it was unfeasible to train larger architectures. Nevertheless, we believe our comparisons between DMLM and its direct competitor, MLM, have been fair, as we have done our best
to set a level playing field between the two. Thus, while we understand that this is a significant limitation in terms of comparability to larger models, we still think the results we have obtained could pave the way for further exploration in this direction. Moreover, we have performed architecture scaling experiments to show that it is important to continue research in this direction, and test DMLM’s capabilities on larger networks, while we did not perform a similar comparison with MLM because several works have already explored how MLM scales with network size (Turc et al., 2019).

Applying DMLM only half of the time Although we acknowledge that our choice to apply DMLM to only half of the sentences can be seen as arbitrary, we argue that it is a sound choice given the nature of our objective. Indeed, we did not want our models to rely too much on the definitions provided, or they would have required them at inference time. Such a requirement is mostly unfeasible, as it would demand running a WSD pipeline before the model’s inference, and this is incompatible or unnecessary with most downstream settings. Nevertheless, we plan on training other architectures with different frequencies, so as to better assess how impactful this hyperparameter is.

Training corpus domain Our models are trained on a sense-tagged version of WikiText-103, which only contains text coming from Wikipedia, and thus is very descriptive in style. While many other works have based their pre-training corpora on Wikipedia, we do recognize that this might be a limitation, especially for downstream tasks.

Training on longer sequences In this work, we trained language models on sentences, as opposed to what is commonly done in the literature, i.e., longer sequences of text which are usually concatenated sentences. We see a limitation here in that, in its current formulation, DMLM does not support training on longer sequences as we have no way of discerning between multiple definitions appended to our input sequence. Nonetheless, while we performed WSD at the sentence level, the corpus can be brought back to full documents, which would make sequence-level training feasible with the available data, provided that an extension to DMLM that supports multiple definitions is designed. We leave such an extension to future work.

Scaling to multiple languages Our formulation of Descriptive Masked Language Modeling can be applied to, as far as we know, virtually any language. Moreover, we argue that it might be possible, in a multilingual setting, that definitions of the same sense could help in aligning the output representations of the trained models for words sharing the same sense. Nevertheless, having said this, it is worth noting that there might be two impediments to achieving multilinguality. First, in our work, we leveraged English Word Sense Disambiguation, which, despite its recent advancements, is still far from performing the task equally well on other, even high-resource, languages (cf. Pasini et al. (2021)). Second, we decided to employ definitions coming from sense inventories which, at least in English, cover a wide number of senses with meaningful descriptions, but this might not be the case for other languages, especially low-resource or endangered ones, with BabelNet (Navigli et al., 2021) being the largest resource providing textual definitions in hundreds of languages.

Reproducibility We acknowledge that, even by releasing the code and dataset on which our models are trained, it might be hard for other interested entities (e.g., groups, people, institutions) to reproduce this work, as our training runs lasted up to 8.5 days on our multi-GPU setup.

8 Acknowledgements

The authors gratefully acknowledge the support of the ERC Consolidator Grant MOUSSE No. 726487, of the PNRR MUR project PE0000013-FAIR, and the PerLIR project (Personal Linguistic resources in Information Retrieval) funded by the MIUR Progetti di ricerca di Rilevante Interesse Nazionale programme (PRIN 2017).

References


Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In ICLR.


Jan Hajic, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Márquez, Adam Meyers, Joakim Nivre, Sebastian


Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020a. On the variance of the adaptive learning rate and beyond. In International Conference on Learning Representations.


<table>
<thead>
<tr>
<th>Baselines</th>
<th>C09</th>
<th>C09OOD</th>
<th>C12</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT small</td>
<td>89.4</td>
<td>80.8</td>
<td>81.4</td>
</tr>
<tr>
<td>TinyBERT</td>
<td>90.8</td>
<td>82.3</td>
<td>84.1</td>
</tr>
<tr>
<td>DistilBERT</td>
<td>91.1</td>
<td>82.8</td>
<td>84.9</td>
</tr>
<tr>
<td>BERT base</td>
<td>91.7</td>
<td>84.0</td>
<td>85.2</td>
</tr>
<tr>
<td>BERT large</td>
<td><strong>92.4</strong></td>
<td><strong>85.6</strong></td>
<td><strong>86.6</strong></td>
</tr>
<tr>
<td>DistilBERTDEF</td>
<td>91.2</td>
<td>82.8</td>
<td>84.9</td>
</tr>
<tr>
<td>BERTDEF base</td>
<td>91.8</td>
<td>84.3</td>
<td>85.4</td>
</tr>
<tr>
<td>BERTDEF large</td>
<td><strong>92.4</strong></td>
<td><strong>85.5</strong></td>
<td><strong>86.6</strong></td>
</tr>
</tbody>
</table>

Results On GLUE (Table 6), both encoder-decoder models achieve scores that are directly comparable to their encoder-only counterparts (MLM_{43M} and DMLM_{43M}), and for which the same conclusions can be drawn, i.e., DMLM-enhanced models achieve far better results on semantic tasks.

On SRL (Table 5), the same point stands, with the DMLM models surpassing their MLM-only counterparts in every benchmark, a finding consistent with that of encoder-only models.

In conclusion, we have seen how DMLM can be applied effectively to an encoder-decoder architecture as well as to an encoder-only architecture, with consistent gains over MLM-only pre-training.

### B WSD System for Corpus Tagging

In this section, we describe the experimental setup in which we trained our Word Sense Disambiguation system.

#### B.1 Preliminaries

In Word Sense Disambiguation (WSD), models are required to choose, given an ambiguous word in some context, the most appropriate meaning the word takes on among the set of its possible meanings. For example, in the sentence “the mouse ate the cheese”, the word *mouse* is encountered in its meaning of *animal*, and not in its meaning of *device*.

Thus, we need a way to link words and meanings, or concepts, such that it is possible to obtain all the possible concepts of a given word, and to obtain all the lexicalizations of a given concept. These “links” are usually provided by a sense inventory

11We favor the encoder, in terms of number of layers, as we do not deal with generative tasks.

12We used the tokenizer of the bert-base-cased model available from the HuggingFace Transformers library (Wolf et al., 2020).
which, given some word (and its part-of-speech tag), returns all its possible concepts.

An example of a commonly used sense inventory is WordNet, where concepts are called synsets, i.e., sets of synonyms representing the same meaning, and where a sense represents a (word, synset) pair, i.e., a word in one of its possible meanings. Moreover, WordNet synsets are semantically rich units that, aside from their possible lexicalizations, also contain, for example, relations to other synsets, such as hypernyms, hyponyms, meronyms, among others; furthermore, synsets are also associated with a natural language definition that describes the meaning they represent.

In DMLM, since we need natural language descriptions of given words in contexts, we disambiguate and retrieve the definition associated with the chosen meaning.

### Table 6: Results on GLUE and WiC. For GLUE, per-task metrics are reported in Appendix D; MRPC and QQP both display F1 / accuracy, while STS-B shows Pearson / Spearman Rank. Underlined task names represent those with repeated runs.

<table>
<thead>
<tr>
<th>Inventory</th>
<th>Instances</th>
<th>Senses</th>
<th>Synsets</th>
<th>Train Synsets</th>
</tr>
</thead>
<tbody>
<tr>
<td>WordNet</td>
<td>243,289</td>
<td>206,941</td>
<td>117,659</td>
<td>25,913</td>
</tr>
<tr>
<td>Wiktionary</td>
<td>66,570</td>
<td>752,473</td>
<td>677,465</td>
<td>60,482</td>
</tr>
<tr>
<td>ODE</td>
<td>785,551</td>
<td>94,341</td>
<td>79,004</td>
<td>78,105</td>
</tr>
</tbody>
</table>

### Table 7: Statistic by sense inventory. Instances describes the number of annotated instances comprising of the training, validation and test set available for each inventory. Columns Senses and Synsets show the number of total senses and total synsets contained in each inventory. Column Train Synsets instead, shows the number of synsets that can be found in the training sets of each inventory, respectively.

<table>
<thead>
<tr>
<th>Inventories</th>
<th>Wn</th>
<th>Ox</th>
<th>Wk</th>
<th>MFS</th>
<th>LFS</th>
<th>UnS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wn</td>
<td>80.7</td>
<td>67.9</td>
<td>—</td>
<td>93.7</td>
<td>55.7</td>
<td>76.8</td>
</tr>
<tr>
<td>Ox</td>
<td>70.3</td>
<td>86.3</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Wn + Ox</td>
<td>81.5</td>
<td>87.7</td>
<td>—</td>
<td>94.0</td>
<td>57.9</td>
<td>77.5</td>
</tr>
<tr>
<td>Wn + Ox + Wk</td>
<td>80.1</td>
<td>87.2</td>
<td>86.2</td>
<td>90.4</td>
<td>58.6</td>
<td>81.3</td>
</tr>
</tbody>
</table>

### B.2 WSD Model Details & Evaluation

Table 7 reports various statistics on the sense inventories, namely WordNet, Oxford and Wiktionary. We can observe that both the amount of data available for training and the number of senses differ greatly between each inventory.

Following Barba et al. (2021), we use as underlying architecture BART_large and train the model to extract the correct definition among those given as input to the model. We follow the hyperparameters of the reference paper but train the model on all the inventories together, our model achieves performances comparable to the original ones. While the results on the test set suggest that including
We note that this drop is only due to the Most Frequent Senses classification. Indeed, performing the same analysis introduced in the original paper, we tested our model on three different partitions of the WordNet test set:

- **MFS**, containing all the instances in the test set annotated with the sense that is the most frequent for the target word in the training set;

- **LFS**, containing all the instances in the test set annotated with a sense that is not the most frequent for the target word in the training set, but does appear in the training set;

- **Unseen Synset**, containing all the instances in the test set annotated with a synset that is not in the training set.

Since we were classifying a big corpus with possibly many senses that were unseen or rare during training we preferred to have a model with a classification less biased towards the most frequent senses seen during training.

### C 1-NN WSD Experiment Details

As stated in the paper, to evaluate the performances of the systems for Word Sense Disambiguation we followed the experimental setting of Wiedemann et al. (2019); Loureiro et al. (2022). However, in contrast to Wiedemann et al. (2019) and Loureiro et al. (2022), we disregarded the MFS (Most Frequent Sense) fallback policy. Specifically, this policy consists of the following procedure: at test time, whenever a word to disambiguate is not present in the training set (i.e., SemCor), the first sense for that word in WordNet is predicted. In our setting, on the other hand, words that are not present in SemCor are automatically marked as wrong. Our reasoning for this choice was the following: given that our intention was to compare the output spaces of our models, we felt it unnecessary to include such a virtual enhancement, as it would benefit every system in the same way.

### D GLUE Details

We provide additional details about GLUE tasks here, and dataset sizes in Table 9. Regarding metrics, unless specified differently, the reported score is an accuracy value.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train</th>
<th>Validation</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoLA</td>
<td>8,551</td>
<td>1,043</td>
<td>1,063</td>
</tr>
<tr>
<td>SST2</td>
<td>67,349</td>
<td>872</td>
<td>1,821</td>
</tr>
<tr>
<td>MRPC</td>
<td>3,668</td>
<td>408</td>
<td>1,725</td>
</tr>
<tr>
<td>STSB</td>
<td>5,749</td>
<td>1,400</td>
<td>1,379</td>
</tr>
<tr>
<td>QQP</td>
<td>363,846</td>
<td>40,430</td>
<td>390,965</td>
</tr>
<tr>
<td>MNLI(_m)</td>
<td>392,702</td>
<td>9,815</td>
<td>9,796</td>
</tr>
<tr>
<td>MNLI(_mm)</td>
<td>/</td>
<td>9,832</td>
<td>9,847</td>
</tr>
<tr>
<td>QNLI</td>
<td>104,743</td>
<td>5,463</td>
<td>5,463</td>
</tr>
<tr>
<td>RTE</td>
<td>2,490</td>
<td>277</td>
<td>3,000</td>
</tr>
</tbody>
</table>

Table 9: GLUE corpus statistics.

**CoLA** Corpus of Linguistic Acceptability (Warstadt et al., 2019). The task is to determine whether a given sentence is grammatical or not. Results report Matthew’s Correlation (Matthews, 1975).

**SST-2** Stanford Sentiment Treebank (Socher et al., 2013). The task is to determine if the sentence is positive or negative in sentiment.

**MRPC** Microsoft Research Paraphrase Corpus (Dolan and Brockett, 2005). The task is to predict whether two sentences are semantically equivalent or not. Results report F1 / accuracy.

**STS-B** Semantic Textual Similarity (Cer et al., 2017). The task is to predict how semantically similar two sentences are on a scale of 1 to 5. Results report Pearson Correlation (Pearson, 1895) and Spearman’s Rank Correlation (Spearman, 1904).

**QQP** Quora Question Pairs. The task is to determine whether a pair of questions are semantically equivalent. Results report F1 / accuracy.

**MNLI** Multi-genre Natural Language Inference (Williams et al., 2018). Given a premise sentence and a hypothesis sentence, the task is to predict whether the premise entails the hypothesis, contradicts the hypothesis, or neither.

**QNLI** Question Natural Language Inference; base on SQuAD (Rajpurkar et al., 2016). The task is to predict whether a context sentence contains the answer to a question.

**RTE** Recognizing Textual Entailment (Giampiccolo et al., 2007). Given a premise sentence and a hypothesis sentence, the task is to predict whether the premise entails the hypothesis or not.
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?

Appendices A-E

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?

5.2-5.5

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.)?

Not applicable. 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.?

Not applicable. Left blank.

D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants’ demographic (e.g., country of residence)?

Not applicable. Left blank.

D3. Did you discuss whether and how consent was obtained from people whose data you’re using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

Not applicable. Left blank.

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

Not applicable. Left blank.

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

Not applicable. Left blank.