DMLM: Descriptive Masked Language Modeling

Edoardo Barba[1] **Niccolò Campolungo**[1,2*] **Roberto Navigli**[1]
[1] Sapienza NLP Group, Sapienza University of Rome
[2] Sunglasses.ai
{barba,navigli}@diag.uniroma1.it
niccolo.campolungo@sunglasses.ai

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

*Work carried out during the PhD programme at the Sapienza University of Rome.

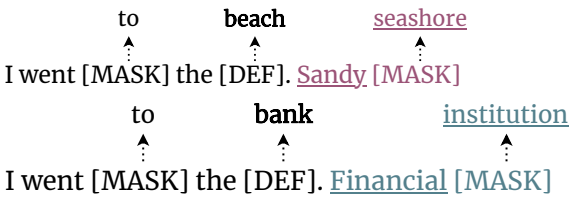


Figure 1: Two examples of DMLM, where a model has to predict a described masked word while simultaneously performing standard MLM.

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[1] 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, `[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 `[MASK]` ." as input, and would be asked to predict *beach* from its corresponding `[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., `[DEF]`, and, iii) it appends d^{w_i} to our input sequence after a special `[DEFINE]` token. Going back to our example, we would obtain "I went to the `[DEF]`. `[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 `[MASK]` the `[DEF]`. `[DEFINE]` Sandy `[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 contextualize both the `[MASK]` words and the special `[DEF]` word;
- The model, especially in ambiguous contexts, must exploit the definition to predict the `[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[2] is split into subwords, we replicate the `[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.

Inventory	Tokens	Instances	Token Vocab.	Definitions
WordNet	98 M	44 M	224 K	89 K
ODE	98 M	72 M	84 K	79 K
Wiktionary	98 M	69 M	183 K	91 K

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.

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**,[3] 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 $\text{BERT}_{\text{base}}$ and around 24h for $\text{BERT}_{\text{large}}$. 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

[3] https://en.wiktionary.org/

[4] A more complete description of the sense inventories, the corpora used for training, evaluating and testing the WSD system, as well as a breakdown of its performances, can be found in Appendix B.

[5] Factoring 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.[6]

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 $\text{BERT}_{\text{small}}$ (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 $\text{BERT}_{\text{base}}$ and $\text{BERT}_{\text{large}}$ (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.

[6] For our sense-tagged WikiText-103 corpus, we remove around 17.3% of all sentences.

		#Params	CoLA	SST-2	MRPC	STS-B	QQP	MNLI_{M}	MNLI_{MM}	QNLI	RTE
Baselines	$\text{BERT}_{\text{base}}$	110M	58.57	92.43	88.89 / 84.31	88.91 / 88.70	87.29 / 90.57	83.42	84.08	90.92	67.15
	$\text{BERT}_{\text{large}}$	335M	62.58	93.46	90.36 / 86.03	90.40 / 90.33	87.96 / 91.04	85.80	85.84	92.28	71.48
	$\text{BERT}_{\text{small}}$	29M	30.78	87.39	85.03 / 76.96	86.33 / 86.32	84.27 / 87.85	76.69	36.76	86.49	62.45
	DistilBERT	66M	47.45	90.48	87.32 / 82.35	84.54 / 84.39	85.40 / 89.14	80.47	81.66	87.24	57.76
	TinyBERT	66M	44.28	91.51	90.39 / 86.76	89.30 / 89.18	86.55 / 89.94	83.78	70.97	90.39	61.01
Our work	MLM_{43M}	43M	**27.02**	89.22	79.53 / 70.10	41.27 / 38.40	79.40 / 84.33	72.56	74.08	76.61	54.87
	DMLM_{43M}	43M	26.51	**89.57**	**86.35 / 82.31**	**86.15 / 86.12**	**85.26 / 88.13**	**80.29**	**79.51**	**85.16**	**62.31**
	DMLM_{20M}	20M	27.07	87.54	83.61 / 74.43	85.38 / 85.41	83.94 / 87.14	78.12	75.12	83.37	59.54
	DMLM_{66M}	66M	**53.13**	**91.92**	**90.51 / 86.89**	**89.73 / 89.64**	**87.72 / 90.85**	**84.65**	**83.92**	**89.48**	**65.69**

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.

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_{M}, MNLI_{MM} 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.

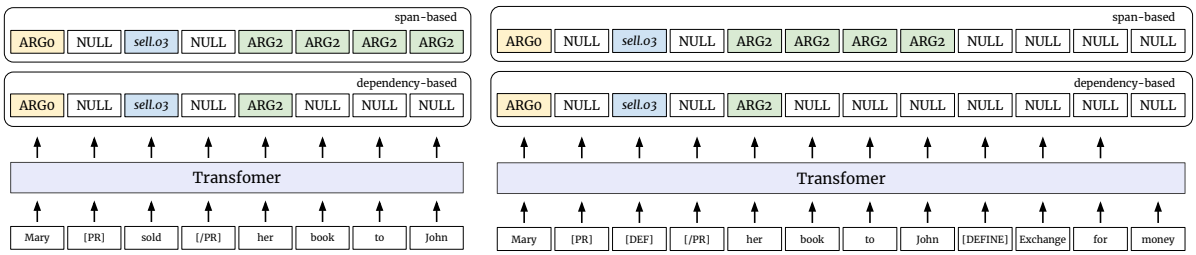
DMLM scaling On the one hand, despite the difference in size, we observe similar performances between our 20M model and $\text{BERT}_{\text{small}}$ (50% larger). Interestingly, the only task where our model strongly outperforms $\text{BERT}_{\text{small}}$ is MNLI_{MM}, with a 40 points difference,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

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 $\text{BERT}_{\text{small}}$ in MNLI, with the MNLI_{MM} lagging behind MNLI_{M} 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_{MM} ($75.12_{\text{20M}} \rightarrow 79.51_{\text{43M}} \rightarrow 83.92_{\text{66M}}$), justifying future research efforts in scaling up network sizes.

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 and Navigli (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.



(a) Without definitions

(b) With definitions

Figure 2: SRL transformer input with and without definitions included.

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 $5 * 10^{-5}$ 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[7] 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.[8]

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., $\text{BERT}_{\text{large}}$ for Conia and Navigli (2020); Shi and Lin (2019).

Regarding our models, DMLM_{43M} consistently outperforms MLM_{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_{66M} comfortably beats both DistilBERT and TinyBERT, and almost achieves scores around $\text{BERT}_{\text{base}}$, 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.[9] As shown

	Models	C09	C09_{OOD}	C12
Baselines	$\text{BERT}_{\text{small}}$	89.4	80.8	81.4
	TinyBERT	90.8	82.3	84.1
	DistilBERT	91.1	82.8	84.9
	$\text{BERT}_{\text{base}}$	91.7	84.0	85.2
	$\text{BERT}_{\text{large}}$	**92.4**	**85.6**	**86.6**
	$\text{DistilBERT}^{\text{DEF}}$	91.2	82.8	84.9
	$\text{BERT}^{\text{DEF}}_{\text{base}}$	91.8	84.3	85.4
	$\text{BERT}^{\text{DEF}}_{\text{large}}$	**92.4**	85.5	**86.6**
SoTA	Shi and Lin (2019)	92.4	—	86.5
	Conia and Navigli (2020)	**92.6**	**85.9**	**87.3**
	Blloshmi et al. (2021)	92.4	85.2	**87.3**
Base	MLM_{43M}	89.1	79.3	82.1
	$\text{MLM}^{\text{DEF}}_{\text{43M}}$	89.3	79.2	82.3
	DMLM_{43M}	90.4	81.2	83.3
	$\text{DMLM}^{\text{DEF}}_{\text{43M}}$	**91.6**	**83.5**	**84.9**
Scaling	DMLM_{20M}	88.9	79.7	82.0
	$\text{DMLM}^{\text{DEF}}_{\text{20M}}$	90.0	81.2	82.5
	DMLM_{66M}	91.7	83.6	85.1
	$\text{DMLM}^{\text{DEF}}_{\text{66M}}$	**92.2**	**84.8**	**85.9**

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_{OOD} represents the Out-Of-Domain portion of the CoNLL-2009 test set.

[7] https://ufal.mff.cuni.cz/conll2009-st/scorer.html

[8] https://www.cs.upc.edu/~srlconll/soft.html

[9] 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.

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 $\text{BERT}_{\text{base}}$, 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 $\text{BERT}_{\text{large}}$, 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, DMLM_{43M} surpasses MLM_{43M} by 3.6 F1 points; interestingly, we perform in the same ballpark as $\text{BERT}_{\text{large}}$ both with DMLM_{43M} and DMLM_{20M}, despite the latter being $\frac{1}{16}$th 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 $\text{DMLM}^{\text{DEF}}_{\text{66M}}$ achieving, in this setting, 5.4 F1 points more than $\text{BERT}_{\text{large}}$ 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

[10] We report a detailed explanation of the Word Sense Disambiguation experimental setting in Section C.

		WSD	Cosine Similarities		
		ALL↑	Sense	Lemma-PoS (Δ)	Random
Baselines	$\text{BERT}_{\text{small}}$	59.3	82.9	82.8 (+0.1)	60.6
	TinyBERT	58.7	82.2	83.1 (-0.9)	61.6
	DistilBERT	60.9	85.0	85.4 (-0.4)	62.9
	$\text{BERT}_{\text{base}}$	61.8	76.3	76.5 (-0.2)	48.6
	$\text{BERT}_{\text{large}}$	**62.9**	60.9	59.9 (+1.0)	18.0
Base	MLM_{43M}	59.6	56.1	58.4 (-2.3)	28.0
	DMLM_{43M}	63.2	63.1	61.6 (+1.5)	19.1
	$\text{DMLM}^{\text{DEF}}_{\text{43M}}$	**65.7**	83.9	65.9 (+18.0)	31.9
Scaling	DMLM_{20M}	62.5	67.3	66.4 (+0.9)	22.3
	DMLM_{66M}	64.1	65.4	60.4 (+5.0)	15.7
	$\text{DMLM}^{\text{DEF}}_{\text{20M}}$	64.1	75.8	68.9 (+6.9)	43.2
	$\text{DMLM}^{\text{DEF}}_{\text{66M}}$	**68.3**	82.4	60.2 (+22.2)	34.3

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.

that share the same lemma and Part-of-Speech tag, regardless of their meaning, and iii) a random set of 50,000 pairs of contextualized words as reference.

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 $\text{BERT}_{\text{large}}$.

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 DMLM objective does not explicitly favor it. Second, the output space shows around the same clustering behavior as $\text{BERT}_{\text{large}}$, 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

Edoardo Barba, Tommaso Pasini, and Roberto Navigli. 2021. ESC: Redesigning WSD with extractive sense comprehension. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4661–4672, Online. Association for Computational Linguistics.

Michele Bevilacqua, Marco Maru, and Roberto Navigli. 2020. Generationary or "how we went beyond word sense inventories and learned to gloss". In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7207–7221, Online. Association for Computational Linguistics.

Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Recent trends in word sense disambiguation: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4330–4338. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in NLI: Ways (not) to go beyond simple heuristics. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 125–135, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rexhina Blloshmi, Simone Conia, Rocco Tripodi, and Roberto Navigli. 2021. Generating senses and roles: An end-to-end model for dependency- and span-based semantic role labeling. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3786–3793. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Niccolò Campolungo, Federico Martelli, Francesco Saina, and Roberto Navigli. 2022. DiBiMT: A novel benchmark for measuring Word Sense Disambiguation biases in Machine Translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4331–4352, Dublin, Ireland. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Ting-Yun Chang, Ta-Chung Chi, Shang-Chi Tsai, and Yun-Nung Chen. 2018. xsense: Learning sense-separated sparse representations and textual definitions for explainable word sense networks. *arXiv preprint arXiv:1809.03348*.

Qianglong Chen, Feng-Lin Li, Guohai Xu, Ming Yan, Ji Zhang, and Yin Zhang. 2022. Dictbert: Dictionary description knowledge enhanced language model pre-training via contrastive learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4086–4092. International Joint Conferences on Artificial Intelligence Organization. Main Track.

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*.

Simone Conia and Roberto Navigli. 2020. Bridging the gap in multilingual semantic role labeling: a language-agnostic approach. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1396–1410, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew W. Senior, Paul A. Tucker, Ke Yang, and Andrew Y. Ng. 2012. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1232–1240.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

Daniel Gildea and Daniel Jurafsky. 2000. Automatic labeling of semantic roles. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 512–520, Hong Kong. Association for Computational Linguistics.

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

Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado. Association for Computational Linguistics.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. SenseBERT: Driving some sense into BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4656–4667, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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*.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020b. K-bert: Enabling language representation with knowledge graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):2901–2908.

Daniel Loureiro, Alípio Mário Jorge, and Jose Camacho-Collados. 2022. Lmms reloaded: Transformer-based sense embeddings for disambiguation and beyond. *Artificial Intelligence*, 305:103661.

Daniel Loureiro, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2021. Analysis and evaluation of language models for word sense disambiguation. *Computational Linguistics*, 47(2):387–443.

B.W. Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

George A. Miller, Richard Beckwith, Christiane D. Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):10:1–10:69.

Roberto Navigli, Michele Bevilacqua, Simone Conia, Dario Montagnini, and Francesco Cecconi. 2021. Ten years of babelnet: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4559–4567. ijcai.org.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. XL-WSD: an extra-large and cross-lingual evaluation framework for word sense disambiguation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13648–13656. AAAI Press.

Karl Pearson. 1895. Note on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London Series I*, 58:240–242.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Jason Phang, Phil Yeres, Jesse Swanson, Haokun Liu, Ian F. Tenney, Phu Mon Htut, Clara Vania, Alex Wang, and Samuel R. Bowman. 2020. `jiant` 2.0: A software toolkit for research on general-purpose text understanding models. `http://jiant.info/`.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *CoRR*, abs/2003.08271.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, Valencia, Spain. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020a. Sensembert: Context-enhanced sense embeddings for multilingual word sense disambiguation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8758–8765. AAAI Press.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020b. With more contexts comes better performance: Contextualized sense embeddings for all-round word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3528–3539. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple BERT models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936. PMLR.

Charles Spearman. 1904. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101.

Wilson L. Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism & Mass Communication Quarterly*, 30:415 – 433.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does BERT make any sense? interpretable word sense disambiguation with contextualized embeddings. *CoRR*, abs/1909.10430.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

Wenhao Yu, Chenguang Zhu, Yuwei Fang, Donghan Yu, Shuohang Wang, Yichong Xu, Michael Zeng, and Meng Jiang. 2022. Dict-BERT: Enhancing language model pre-training with dictionary. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1907–1918, Dublin, Ireland. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

	Models	C09	$C09_{OOD}$	C12
Baselines	$BERT_{small}$	89.4	80.8	81.4
	TinyBERT	90.8	82.3	84.1
	DistilBERT	91.1	82.8	84.9
	$BERT_{base}$	91.7	84.0	85.2
	$BERT_{large}$	**92.4**	**85.6**	**86.6**
	$DistilBERT^{DEF}$	91.2	82.8	84.9
	$BERT^{DEF}_{base}$	91.8	84.3	85.4
	$BERT^{DEF}_{large}$	**92.4**	85.5	**86.6**
SoTA	Shi and Lin (2019)	92.4	—	86.5
	Conia and Navigli (2020)	**92.6**	**85.9**	**87.3**
	Blloshmi et al. (2021)	92.4	85.2	**87.3**
Our work	MLM_{43M}	89.1	79.3	82.1
	MLM^{DEF}_{43M}	89.3	79.2	82.3
	$DMLM_{43M}$	90.4	81.2	83.3
	$DMLM^{DEF}_{43M}$	**91.6**	**83.5**	**84.9**
	$DMLM_{20M}$	88.9	79.7	82.0
	$DMLM^{DEF}_{20M}$	90.0	81.2	82.5
	$DMLM_{66M}$	91.7	83.6	85.1
	$DMLM^{DEF}_{66M}$	**92.2**	**84.8**	**85.9**
	MLM_{ED}	88.8	79.0	82.0
	MLM^{DEF}_{ED}	89.1	79.0	82.2
	$DMLM_{ED}$	90.3	81.3	83.1
	$DMLM^{DEF}_{ED}$	91.5	**83.5**	**85.0**

Table 5: 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; $C09_{OOD}$ represents the Out-Of-Domain portion of the CoNLL-2009 test set. This table is identical to Table 3, with the addition of the last four rows regarding the encoder-decoder version of DMLM.

A DMLM in Encoder-Decoder Architectures

DMLM's formulation can easily be applied to encoder-decoder architectures, with very small adjustments: following Lewis et al. (2020), we, i) do not replicate `[DEF]` tokens in the input sequence even when they would be split into multiple subwords by the underlying tokenizer, and, ii) ask the model to generate the whole input sequence, definition included. Thus, given our running example (Figure 1), the model would see "I went `[MASK]` the `[DEF]`. `[DEFINE]` Sandy `[MASK]`" as input and would be asked to predict "I went to the beach . `[DEFINE]` Sandy seashore".

Architecture We used the architecture of Lewis et al. (2020), with the same hidden size and attention heads as our encoder-only module, but with 4 encoder layers and 2 decoder layers.[11]

Moreover, to make the encoder-decoder and the encoder-only architectures as comparable as possible, and to maintain a similar size, we trained our encoder-decoder model using the same vocabulary and tokenizer of our encoder-only model,[12] thus totaling around 46M parameters.

Similarly to MLM_{43M} and $DMLM_{43M}$, we trained two 46M encoder-decoder architectures with and without DMLM, which we dubbed $DMLM_{ED}$ and MLM_{ED}, respectively.

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

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

[12] We used the tokenizer of the `bert-base-cased` model available from the HuggingFace Transformers library (Wolf et al., 2020).

		#Params	CoLA	SST-2	MRPC	STS-B	QQP	$MNLI_M$	$MNLI_{MM}$	QNLI	RTE
Literature	DistilBERT (Sanh et al., 2019)	66M	51.30	91.30	87.50	86.90	88.50	82.20	—	89.20	59.90
	SenseBERT† (Levine et al., 2020)	110M	54.60	92.20	89.20 / 85.20	83.50 / 82.30	70.30 / 88.80	83.60	—	90.60	67.50
	Dict-BERT (Yu et al., 2022)	110M	61.68	92.65	87.21	89.68	90.81	84.34	—	91.20	72.89
	BERT (from Clark et al., 2020)	335M	60.60	93.20	88.00	90.00	91.30	86.60	—	92.30	70.40
	ELECTRA (Clark et al., 2020)	335M	69.10	96.90	90.80	92.60	92.40	90.90	—	95.00	88.00
	SpanBERT (Joshi et al., 2020)	335M	64.30	94.80	90.90 / 87.90	89.90 / 89.10	71.90 / 89.50	88.10	87.70	94.30	79.00
	DictBERT (Chen et al., 2022)	335M	68.60	97.80	93.20	92.10	90.80	91.10	—	96.80	89.40
	BART (Lewis et al., 2020)	406M	62.80	96.60	90.40	91.20	92.50	89.90	90.10	94.90	87.00
Baselines	$BERT_{base}$	110M	58.57	92.43	88.89 / 84.31	88.91 / 88.70	87.29 / 90.57	83.42	84.08	90.92	67.15
	$BERT_{large}$	335M	62.58	93.46	90.36 / 86.03	90.40 / 90.33	87.96 / 91.04	85.80	85.84	92.28	71.48
	$BERT_{small}$	29M	30.78	87.39	85.03 / 76.96	86.33 / 86.32	84.27 / 87.85	76.69	36.76	86.49	62.45
	DistilBERT	66M	47.45	90.48	87.32 / 82.35	84.54 / 84.39	85.40 / 89.14	80.47	81.66	87.24	57.76
	TinyBERT	66M	44.28	91.51	90.39 / 86.76	89.30 / 89.18	86.55 / 89.94	83.78	70.97	90.39	61.01
Our work	MLM_{43M}	43M	**27.02**	89.22	79.53 / 70.10	41.27 / 38.40	79.40 / 84.33	72.56	74.08	76.61	54.87
	$DMLM_{43M}$	43M	26.51	**89.57**	**86.35 / 82.31**	**86.15 / 86.12**	**85.26 / 88.13**	**80.29**	**79.51**	**85.16**	**62.31**
	$DMLM_{20M}$	20M	27.07	87.54	83.61 / 74.43	85.38 / 85.41	83.94 / 87.14	78.12	75.12	83.37	59.54
	$DMLM_{66M}$	66M	**53.13**	**91.92**	**90.51 / 86.89**	**89.73 / 89.64**	**87.72 / 90.85**	**84.65**	**83.92**	**89.48**	**65.69**
	MLM_{ED}	46M	**25.22**	85.16	77.55 / 70.41	43.19 / 38.30	81.62 / 85.49	75.98	75.77	83.15	54.79
	$DMLM_{ED}$	46M	22.22	**87.73**	**84.26 / 76.47**	**81.96 / 81.59**	**84.23 / 88.13**	**76.96**	**78.01**	**83.58**	**58.48**

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. † means that the results are on the GLUE test set, as the ones on the dev set were not available.

Inventory	Instances	Senses	Synsets	Train Synsets
WordNet	233,289	206,941	117,659	25,913
Wiktionary	66,570	752,473	677,465	60,482
ODE	785,551	94,341	79,004	78,105

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.

Inventories	Wn	Ox	Wk	MFS	LFS	UnS
Wn†	80.7	67.9	—	93.7	55.7	76.8
Ox†	70.3	86.3	—	—	—	—
Wn + Ox†	**81.5**	**87.7**	—	**94.0**	57.9	77.5
Wn + Ox + Wk	80.1	87.2	86.2	90.4	**58.6**	**81.3**

Table 8: Word Sense Disambiguation performances of ESCHER trained on different inventories at the same time. *Inventories* shows the inventories used at training time. Columns *Wn*, *Ox*, *Wk* show the performances on the test sets of WordNet, ODE and Wiktionary respectively. Columns *MFS*, *LFS*, *UnS* show the performances of the models on the Most Frequent Senses, Least Frequent Senses and Unseen Synsets respectively. † indicates that the results were taken from the original paper or using model weights made available by the authors.

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.

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 three inventories jointly.

As we can see from Table 8, when trained on all the inventories together, our model achieves performances comparable to the original ones. While the results on the test set suggest that including

Wiktionary in the training deteriorates the performances, 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.

Dataset	Train	Validation	Test
CoLA	8,551	1,043	1,063
SST2	67,349	872	1,821
MRPC	3,668	408	1,725
STSB	5,749	1,400	1,379
QQP	363,846	40,430	390,965
MNLI_M	392,702	9,815	9,796
MNLI_MM	/	9,832	9,847
QNLI	104,743	5,463	5,463
RTE	2,490	277	3,000

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 tasks 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.