Does He Wink or Does He Nod? A Challenging Benchmark for Evaluating Word Understanding of Language Models

Lutfi Kerem Senel and **Hinrich Schütze**
Center for Information and Language Processing (CIS), LMU Munich, Germany
lksenel@gmail.com, inquiries@cislmu.org

Abstract

Recent progress in pretraining language models on large corpora has resulted in large performance gains on many NLP tasks. These large models acquire linguistic knowledge during pretraining, which helps to improve performance on downstream tasks via fine-tuning. To assess what kind of knowledge is acquired, language models are commonly probed by querying them with 'fill in the blank' style cloze questions. Existing probing datasets mainly focus on knowledge about *relations between* words and entities. We introduce WDLMPro (Word Definition Language Model Probing) to *evaluate word understanding directly* using dictionary definitions of words. In our experiments, three popular pretrained language models struggle to match words and their definitions. This indicates that they understand many words poorly and that our new probing task is a difficult challenge that could help guide research on LMs in the future.

1 Introduction

Natural language processing (NLP) has advanced drastically in the last decade with the design of larger and more sophisticated models, availability of larger corpora and increasing computational power. Pretrained word embeddings (Mikolov et al., 2013; Pennington et al., 2014) popularized the use of distributed word representations, which became a fundamental building block for NLP systems. Peters et al. (2018a) introduced LSTM-based deep contextual representations and obtained large performance gains by fine-tuning on tasks after unsupervised pretraining (Radford et al., 2018; Howard and Ruder, 2018). More recently, the attention based transformer architecture was shown to use context more effectively (Vaswani et al., 2017) and several subsequent models achieved state of the art results in many NLP tasks by combining the transformer architecture with unsupervised

pretraining and task specific fine-tuning (Devlin et al., 2019; Liu et al., 2019). Radford et al. (2019) showed that language models can be applied to a variety of tasks without task specific fine tuning. This is demonstrated on a much larger scale by Brown et al. (2020).

Deep models improve performance. However, what they actually learn about language and word meaning is still to a large extent unclear due to their uninterpretable nature. For static word embeddings, researchers used word similarity (Hill et al., 2015) and word analogy (Gladkova et al., 2016) tests to shed light on what information is captured in these dense vector spaces. For language models, a great amount of linguistic knowledge is stored in the model parameters (Peters et al., 2018b). Several studies proposed using 'fill in the blank' type cloze statements to test knowledge learned by these models during unsupervised pretraining. Petroni et al. (2019) proposed the LAMA (LAnguage Model Analysis) probe to test the factual and common sense knowledge stored in language models. Similarly, Schick and Schütze (2020) introduced WNLaMPro (WordNet Language Model Probing) to assess the ability of language models to understand words based on their frequency. In WNLaMPro, cloze style questions are generated based on antonym, hypernym and cohyponym relations among words extracted from WordNet.

The existing probing datasets mainly focus on investigating the knowledge about *relations between* words or entities. However, a more direct way of testing whether a language model understands the meaning of a word is to use its dictionary definition. If a pretrained language model truly understands the meaning of a word, then it should be able to match it with its dictionary definition. Based on this motivation, we introduce the *Word Definition*

synset	definition
a_cappella_singing.n.01	*singing without instrumental accompaniment*
caroling.n.01	singing joyful religious songs (especially at Christmas)
crooning.n.01	singing in a soft low tone
singalong.n.01	informal group singing of popular songs
bel_canto.n.01	a style of operatic singing

Table 1: Five candidates from $\mathcal{G}(t)$ for $t=$ *a_cappella_singing.n.01* and their definitions

	Noun	Verb
# of Synset Groups	51260	8487
Average # of Candidates	50.2	47.7
min / max # of Candidates	5 / 404	5 / 593

Table 2: WDLMPro statistics

Language Model Probing (WDLMPro) dataset;[1] it is a challenging benchmark for testing NLP models for their ability to understand words. WDLMPro is essentially a set of thousands of synset groups; each synset group consists of a target word (with its definition) and its taxonomic sisters (with their definitions). Using taxonomic sisters, rather than random word groups, makes the task more challenging for statistical models that are based on the distributional hypothesis since these words have similar distributional characteristics (Lenci, 2008). We evaluate two masked language models, BERT and RoBERTa, and the auto-regressive model GPT-2 on WDLMPro using two different probing tests: (i) match definition to word (D2W) (ii) match word to definition (W2D). We also provide a baseline using static fastText embeddings (Mikolov et al., 2018). We find that all three language models perform clearly better than the baseline. Nevertheless, they have great difficulty matching words and their definitions, implying a poor understanding of word meaning. This is an important result that could help guide research on LMs in the future.

2 WDLMPro

In this section, we introduce WDLMPro (Word Definition Language Model Probing), a dataset to test how well NLP models can match nouns and verbs with their definitions. We view this as a test of how well the models understand lexical meaning.

2.1 Dataset

WordNet (Miller, 1995) is the basis for constructing WDLMPro. A WordNet *synset* contains a set

of synonyms along with a short definition of the synset. Different senses of polysemous words are represented in different synsets providing disambiguation. WordNet connects synsets with each other via semantic relations.

Based on a *target synset* t and the semantic relation hyponymy $<$, we construct a *synset group* $\mathcal{G}$ for the target as follows.

$$\mathcal{G}(t) = \{x | \exists y : t < y \wedge x < y\}$$

that is, G contains all synsets that are "sister hyponyms" to t with respect to a hypernym of t. $\mathcal{G}(t)$, along with the definitions of the synsets in $\mathcal{G}(t)$, will be used to set up the WDLMPro tasks that require matching of words and definitions. We discard groups $\mathcal{G}(t)$ that have a size of less than 5.

In this study, we focus on nouns and verbs, i.e., we create synset groups $\mathcal{G}$ for the nouns and verbs in WordNet. Table 1 displays five members from $\mathcal{G}(t)$ and their definitions for the target *a_cappella_singing.n.01* (see appx. for the target *beckon.v.01*.) Table 2 shows statistics of the dataset.

2.2 Probing Tests

We define two probing tests that are converses of each other:

- **Match definition to word (D2W).** Given a definition and a set of words, the task is to find the word that the definition defines.

- **Match word to definition (W2D).** Given a word and a set of definitions, the task is to find the definition that defines the word.

Each synset group $\mathcal{G}(t)$ gives rise to one instance of D2W by providing the definition of t, and all words in $\mathcal{G}(t)$. The word from $\mathcal{G}(t)$ that matches the definition has then to be identified. (Note that t is a member of $\mathcal{G}(t)$.) Similarly, each synset group $\mathcal{G}(t)$ gives rise to one instance of W2D by providing t and the definitions of all words in $\mathcal{G}(t)$. The correct definition of t has then to be identified among all definition candidates. Note that WordNet definitions by construction do not contain the word

---

Masked Language Model (MLM)	
Noun	__ is <DEF>
	__ means <DEF>
	__ is defined as <DEF>
Verb	definition of __ is to <DEF>
	to <DEF> is the definition of __
Autoregressive Language Model (ALM)	
Noun	<DEF> is the definition of __
Verb	to <DEF> is the definition of __

Table 3: Patterns used for querying language models for nouns and verbs. <DEF> refers to the definition, __ is the mask or missing word that the language model has to predict.

to be defined. So there are no instances where the two tasks are trivial.

2.2.1 Application to language models

In principle, any NLP model can be tested on D2W and W2D. In this paper, we are particularly interested in testing language models. To this end, we convert the data to a format that is suitable for language models, i.e., to cloze-style questions as shown in Table 3. The basic quantity that allows us to assess the compatibility of a word t and a definition is the probability of t being generated for "__" when the definition is substituted for <DEF>.

More precisely, we compute the probability that the string representation of t is being generated. We will denote the string representation of synset t by $\boldsymbol{t}$. We obtain the string representation by removing the word type and sense information from the name of the synset and replacing underscores with white space. For example, synset *warm_up.v.04* is represented by the string "warm up".

Table 3 shows that we define different templates for masked and autoregressive language models. For the masked language models, we average the prediction scores across patterns before ranking the candidates.

2.3 Baselines

For a masked language model (MLM) M, the probability of a candidate $c \in \mathcal{G}(t)$ on W2D is calculated as:

$$P_M^{\text{W2D}}(c|t) = \prod_{i=1}^{|\boldsymbol{t}|} P(\boldsymbol{t}^i|Q(c,|\boldsymbol{t}|))$$

where $\boldsymbol{t} = [\boldsymbol{t}^1, \boldsymbol{t}^2, ..., \boldsymbol{t}^{|\boldsymbol{t}|}]$ is the tokenization produced by M. $Q(c, |\boldsymbol{t}|)$ is the input query created

from one of the patterns (Table 3) with __ replaced with $|\boldsymbol{t}|$ consecutive mask tokens. For an autoregressive language model (ALM) A, we decompose $P(\boldsymbol{t}^i|Q(c), \boldsymbol{t})$ in the standard way:

$$P_A^{\text{W2D}} = \prod_{i=1}^{|\boldsymbol{t}|} P(\boldsymbol{t}^i|Q(c), \boldsymbol{t}^1, ..., \boldsymbol{t}^{i-1})$$

For D2W, we need to compare, given a definition, the probabilities of different candidate words that are generally of different lengths. To ensure a fair comparison, we follow Xiong et al. (2020). For MLMs, we match the number of mask tokens in an input query to the token count of each candidate. The final score is the average log-probability of the masked tokens:

$$P_M^{\text{D2W}}(c|t) = \frac{1}{|\boldsymbol{c}|} \sum_{i=1}^{|\boldsymbol{c}|} \log P(\boldsymbol{c}^i|Q(t, |\boldsymbol{c}|))$$

For ALMs, we use the probability of the first token:

$$P_A^{\text{D2W}}(c|t) = P(\boldsymbol{c}^1|Q(t))$$

Considering further tokens does not make sense since they are often easily predictable from the first token.

We apply our probing test to two different pre-trained MLMs (BERT and RoBERTa) and one ALM (GPT-2). To investigate the effect of model size on the performance, we experiment with both base and large versions of BERT and RoBERTa along with all four sizes of GPT-2 (small, medium, large, xl). For RoBERTa, we capitalize the first letter of the candidate noun since pretrained RoBERTa models are case sensitive and expect a capital letter at the beginning of a sentence.[2]

In addition to the deep contextual language models, we also provide fastText static word embeddings[3] (Mikolov et al., 2018) as a baseline.[4] For fastText embeddings, we tokenize the candidates and their definitions using the NLTK tokenizer and represent them with their average vector. We rank candidates based on their cosine similarity to the target embedding.

---

[2]Not using capitalization resulted in poor performance for single token target words for D2W.

[3]We use the crawl-300d-2M-subword model from https://fasttext.cc/docs/en/english-vectors.html

[4]A reviewer suggests that it would also be interesting to investigate the performance of supervised approaches, e.g., ranking models. Our main focus here is the lexical knowledge acquired in pretraining, so we leave this for future work.

2.4 Measures

We use two measures: precision at 1 (P@1) and a rank score (RS), both based on a ranked results list, either of words or of definitions. P@1 is the percentage of top-ranked items that is correct. We define RS as follows:

$$\text{RS}(L, k) = \frac{L - k}{L - 1}$$

where $L = |\mathcal{G}(t)|$ is the number of candidates and k is the rank of the correct item, $1 \leq k \leq L$. Table 2 shows that the size of $\mathcal{G}(t)$ is highly variable; in contrast to P@1, RS is less affected by this and the random baseline (cf. Tables 4 and 5) is always 0.5.

3 Results

Tables 4 and 5 present W2D and D2W results for BERT, RoBERTa and GPT-2 along with fastText and random baselines. Language models perform clearly better than both baselines. Larger models perform generally better than smaller ones and RoBERTa consistently outperforms BERT. This might be an indication for the correlation between performance on WDLAMPro and downstream performance. However, further investigation is necessary to show the correlation more clearly. For W2D, best performance is achieved by GPT-2$_{xl}$ for nouns (47.3 P@1, 0.81 RS) and by RoBERTa large for verbs (50.8 P@1, 0.84 RS). Performance on D2W is much lower than for W2D for all models. For nouns, RoBERTa large and GPT-2$_{xl}$ perform similarly (28.8 and 29.8 P@1, 0.70 and 0.73 RS) while RoBERTa large achieves the best results for verbs (38.6 P@1, 0.80 RS). Poor performance on D2W compared to W2D might be due to language models' ability to distinguish different definitions better than individual words since definitions are more informative than individual words. Overall GPT-2 models perform better than masked language models (with the exception of Roberta large for verbs), despite using a single pattern as opposed to the multiple patterns used by masked language models. This might indicate that the ALM objective is better at learning word meaning than the MLM objective.

To investigate the effect of frequency, we stratify words into *rare* (fewer than 10 occurrences), *medium* (10 to 99 occurrences) and *frequent* (100 or more occurrences), based on occurrences in WWC[5] (Westbury Wikipedia Corpus, Shaoul (2010)),

[5]Targets that have more than 3 tokens (based on NLTK tokenization) are taken as rare without counting.

Model	Noun		Verb	
	P@1	RS	P@1	RS
Bert$_b$	35.2	0.74	35.3	0.74
Bert$_l$	35.1	0.73	33.6	0.73
Roberta$_b$	37.1	0.75	42.7	0.79
Roberta$_l$	42.1	0.78	50.8	0.84
GPT-2$_s$	38.7	0.76	45.0	0.80
GPT-2$_m$	41.8	0.77	43.6	0.80
GPT-2$_l$	45.7	0.80	48.4	0.83
GPT-2$_{xl}$	47.3	0.81	48.6	0.83
fastText	22.5	0.66	29.1	0.69
Random	7.6	0.50	7.8	0.50

Table 4: P@1 and rank score (RS) on W2D

Model	Noun		Verb	
	P@1	RS	P@1	RS
Bert$_b$	23.7	0.65	19.3	0.65
Bert$_l$	25.4	0.65	19.3	0.65
Roberta$_b$	25.7	0.67	32.6	0.74
Roberta$_l$	28.8	0.70	38.6	0.80
GPT-2$_s$	23.2	0.68	29.2	0.71
GPT-2$_m$	25.3	0.70	27.8	0.72
GPT-2$_l$	28.4	0.72	31.5	0.74
GPT-2$_{xl}$	29.8	0.73	32.8	0.76
fastText	16.5	0.63	20.3	0.69
Random	7.6	0.50	8.0	0.50

Table 5: P@1 and rank score (RS) on D2W

where we use WWC frequency as a substitute for the models' training corpora. We focus on nouns since most verbs in our dataset are relatively frequent. Table 7 shows that, for W2D, all models have a poor understanding of the meaning of rare and medium words. (See appx. for D2W results.) Even for frequent words, P@1 is never above 55.

We additionally break down the results based on the depth of the synsets in the WordNet hierarchy. Specifically, we investigate the performance of the GPT-2$_{xl}$ model on W2D for WordNet nouns, where we take the depth of a synset group as the length of the shortest path from the target synset to the root synset (i.e., *entity.n.01*). Table 6 shows that performance drops steadily as we go deeper in the hierarchy. Lower levels of the WordNet hierarchy contain many scientific terms and names of (sub)species such as types of cattle (e.g., *cattalo*, *hereford*, *galloway*). These results suggest that even very large LMs lack the knowledge necessary to distinguish these terms.

Depth	# synsets	# cand.	RS	P@1
3–5	2106	110	0.94	62.9
6–8	25,232	53	0.83	49.0
9–11	18,521	45	0.81	46.6
12–14	4473	19	0.74	37.4
15–19	928	13	0.67	31.5

Table 6: RS and P@1 results for GPT-2$_{xl}$ on W2D for nouns from different depths of the WordNet hierarchy. # of candidates, RS and P@1 are given as the average across all synsets within the given depth range.

Model	rare	medium	frequent	all
Bert$_b$	26.0	31.1	40.7	35.2
Bert$_l$	23.6	29.8	42.0	35.1
Roberta$_b$	30.8	34.7	40.7	37.1
Roberta$_l$	33.2	38.7	47.2	42.1
GPT-2$_s$	32.9	35.2	42.6	38.7
GPT-2$_m$	34.4	37.4	46.7	41.8
GPT-2$_l$	37.0	41.4	51.1	45.7
GPT-2$_{xl}$	37.7	42.7	53.3	47.3
Random	6.6	7.0	8.2	7.6

Table 7: P@1 scores on W2D for nouns of different frequency ranges

Analysis. The correct definition of the medium frequency verb 'beckon' is 'signal with the hands or nod'. GPT-2$_{xl}$ predicts 'signal by winking'. The correct definition of the frequent noun 'roleplaying' is 'acting a particular role (as in psychotherapy)' GPT-2$_{xl}$ predicts 'acting the part of a character on stage'. So GPT-2$_{xl}$ understands that beckoning is signaling and that roleplaying is acting, but it has not learned to distinguish between different types of signaling and acting. This points to an important future goal for LMs: they should be developed to gain an understanding of words that goes beyond the current superficial state of the art.

Human performance on WDLAMPro. It is beyond the scope of this paper to evaluate human performance on the entirety of WDLAMPro. However, we provide a comparison with human performance on a small subset to provide an intuition about the difficulty of the task. For each of the two tasks, 20 synset groups that have a maximum of 10 candidates are randomly sampled from WD-LAMPro. Then two native English speakers are asked to rank the candidates. Table 8 displays the average performance of the human participants and the language models on this subset. For both tasks, performance of the best model is comparable to the

Model	W2D		D2W	
	P@1	RS	P@1	RS
Bert$_b$	60.0	0.84	35.0	0.64
Bert$_l$	65.0	0.74	35.0	0.69
Roberta$_b$	50.0	0.78	60.0	0.81
Roberta$_l$	55.0	0.80	45.0	0.69
GPT-2$_s$	35.0	0.69	45.0	0.71
GPT-2$_m$	50.0	0.80	50.0	0.73
GPT-2$_l$	60.0	0.84	45.0	0.75
GPT-2$_{xl}$	50.0	0.76	45.0	0.79
Human	62.5	0.88	57.5	0.77

Table 8: LM and human performance on 20 random samples of WDLAMPro.

average human performance.

Human performance is the upper bound for many NLP tasks. We believe that this is not the case for WDLAMPro: arguably, we should aim for models with an excellent understanding of the meanings of words even if it is better than average human understanding. Knowledge based tasks are an analogous case: we should strive for models that know as many facts as possible even if that performance is above average human performance.

4 Conclusion

We introduced WDLMPro, a probing test that helps analyze how well a model understands word meaning. WDLMPro is complementary to existing probing tests that are about *relations* between words or entities. We evaluated three popular pretrained language models on the W2D (word to definition) and D2W (definition to word) tasks. Our findings show that, despite their remarkable performance on many downstream tasks, these models struggle to match a word and its true definition, suggesting an insufficient understanding of word meaning. Relatively poor performance of these powerful models on WDLMPro can be seen as evidence for the limitations of purely distributional systems and the need for incorporating external knowledge. WDLMPro provides an important evaluation benchmark, encouraging design and training of models with precise word understanding.

References

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

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.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15, San Diego, California. Association for Computational Linguistics.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Alessandro Lenci. 2008. Distributional semantics in linguistic and cognitive research. *Italian journal of linguistics*, 20(1):1–31.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Tomas Mikolov, G.s Corrado, Kai Chen, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proc. of the International Conference on Learning Representations (ICLR)*, pages 1–12.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

George A Miller. 1995. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018b. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. In *Technical Report*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Timo Schick and Hinrich Schütze. 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:8766–8774.

C. & Westbury C. Shaoul. 2010. The westbury lab wikipedia corpus.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2020. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *International Conference on Learning Representations*.

A Appendix

synset	definition
beckon.v.01	*signal with the hands or nod*
applaud.v.01	clap one's hands or shout after performances to indicate approval
bow.v.01	bend one's knee or body, or lower one's head
shrug.v.01	raise one's shoulders to indicate indifference or resignation
exsert.v.01	thrust or extend out
wink.v.01	signal by winking
nod.v.01	express or signify by nodding

Table 9: Seven candidates of $\mathcal{G}(t)$ for $t=$ *beckon.v.01* and their definitions

Model	rare	medium	frequent	all
Bert$_b$	14.7	20.6	28.7	23.7
Bert$_l$	12.0	20.1	33.1	25.4
Roberta$_b$	17.7	24.2	29.5	25.7
Roberta$_l$	17.9	25.8	34.5	28.8
GPT-2$_s$	17.3	20.7	26.7	23.2
GPT-2$_m$	17.0	21.1	30.6	25.3
GPT-2$_l$	19.2	24.3	33.9	28.4
GPT-2$_{xl}$	19.3	24.8	36.3	29.8
Random	6.7	7.1	8.3	7.6

Table 10: P@1 scores on D2W for nouns based on target word frequency.