"Is Whole Word Masking Always Better for Chinese BERT?": Probing on Chinese Grammatical Error Correction

Anonymous ACL submission

Abstract

Whole word masking (WWM), which masks all subwords corresponding to a word at once, makes a better English BERT model (Sennrich et al., 2016). For the Chinese language, however, there is no subword because each token is an atomic character. The meaning of a word in Chinese is different in that a word is a compositional unit consisting of multiple characters. Such difference motivates us to investigate whether WWM leads to better context understanding ability for Chinese BERT. To achieve this, we introduce two probing tasks related to grammatical error correction and ask pretrained models to revise or insert tokens in a masked language modeling manner. We construct a dataset including labels for 19,075 tokens in 10,448 sentences. We train three Chinese BERT models with standard character-level masking (CLM), WWM, and a combination of CLM and WWM, respectively. Our major findings are as follows: First, when one character needs to be inserted or replaced, the model trained with CLM performs the best. Second, when more than one character needs to be handled, WWM is the key to better performance. Finally, when being fine-tuned on sentence-level downstream tasks, models trained with different masking strategies perform comparably.[1]

1 Introduction

BERT (Devlin et al., 2018) is a Transformer-based pretrained model, whose prosperity starts from English language and gradually spreads to many other languages. The original BERT model is trained with character-level masking (CLM). [2] A certain percentage (e.g. 15%) of tokens in the input sequence is masked and the model is learned to predict the masked tokens.

It is helpful to note that a word in the input sequence of BERT can be broken into multiple wordpiece tokens (Wu et al., 2016).[3] For example, the input sentence "`She is undeniably brilliant`" is converted to a wordpiece sequence "`She is un ##deni ##ably brilliant`", where "`##`" is a special prefix added to indicate that the token should be attached to the previous one. In this case the word "`undeniably`" is broken into three wordpieces {"`un`", "`##deni`", "`##ably`"}. In standard masked language modeling, CLM may mask any one of them. In this case, if the token "`##ably`" is masked, it is easier for the model to complete the prediction task because "`un`" and "`##deni`" are informative prompts. To address this, Whole word masking (WWM) masks all three subtokens (i.e., {"`un`", "`##deni`", "`##ably`"}) within a word at once. For Chinese, however, each token is an atomic character that cannot be broken into smaller pieces. Many Chinese words are compounds that consisting of multiple characters (Wood and Connelly, 2009). [4] For example, "手机" (`cellphone`) is a word consisting of two characters "手" (`hand`) and "机" (`machine`). Here, learning with WWM would lose the association among characters corresponding to a word.

In this work, we introduce two probing tasks to study Chinese BERT model's ability on character-level understanding. The first probing task is character replacement. Given a sentence and a position where the corresponding character is erroneous, the task is to replace the erroneous character with the correct one. The second probing task is character insertion. Given a sentence and the positions where

[1] We will release the dataset and pretrained models for future research.

[2] Next sentence prediction is the other pretraining task adopted in the original BERT paper. However, it is removed in some following works like RoBERTa (Liu et al., 2019). We do not consider the next sentence prediction in this work.

[3] In this work, wordpiece and subword are interchangeable.

[4] When we describe Chinese tokens, "character" means 字 that is the atomic unit and "word" means 词 that may consist of multiple characters.

a given number of characters should be inserted, the task is to insert the correct characters. We leverage the benchmark dataset on grammatical error correction (Rao et al., 2020a) and create a dataset including labels for 19,075 tokens in 10,448 sentences.

We train three baseline models based on the same text corpus of 80B characters using CLM, WWM, and both CLM and WWM, separately. We have the following major findings. (1) When one character needs to be inserted or replaced, the model trained with CLM performs the best. Moreover, the model initialized from RoBERTa (Cui et al., 2019) and trained with WWM gets worse gradually with more training steps. (2) When more than one character needs to be handled, WWM is the key to better performance. (3) When evaluating sentence-level downstream tasks, the impact of these masking strategies is minimal and the model trained with them performs comparably.

2 Our Probing Tasks

In this work, we present two probing tasks with the goal of diagnosing the language understanding ability of Chinese BERT models. We present the tasks and dataset in this section.

The first probing task is character replacement, which is a subtask of grammatical error correction. Given a sentence $s = \{x_1, x_2, ..., x_i, ..., x_n\}$ of n characters and an erroneous span $es = [i, i+1, ..., i+k]$ of k characters, the task is to replace es with a new span of k characters.

The second probing task is character insertion, which is also a subtask of grammatical error correction. Given a sentence $s = \{x_1, x_2, ..., x_i, ..., x_n\}$ of n characters, a position i, and a fixed number k, the task is to insert a span of k characters between the index i and $i+1$.

We provide two examples of these two probing tasks with $k = 1$ in Figure 1. For the character replacement task, the original meaning of the sentence is "*these are all my ideas*". Due to the misuse of a character at the 7th position, its meaning changed significantly to "*these are all my attention*". Our character replacement task is to replace the misused character "主" with "注". For the character insertion task, what the writer wants to express is "*Human is the most important factor*. However, due to the lack of one character between the 5th and 6th position, its meaning changed to "*Human is the heaviest factor*". The task is to

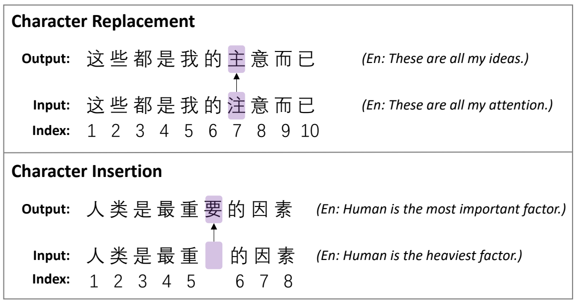

Figure 1: Illustrative examples of two probing tasks. For character replacement (upper box), the highlighted character at 7th position should be replaced with another one. For character insertion (bottom box), one character should be inserted after the 5th position. Translations in English are given in parentheses.

insert "要" after the 5th position. Both tasks are also extended to multiple characters (i.e., $k \geq 2$). Examples can be found at Section 3.2.

We build a dataset based on the benchmark of Chinese Grammatical Error Diagnosis (CGED) in years of 2016, 2017, 2018 and 2020 (Lee et al., 2016; Rao et al., 2017, 2018, 2020b). The task of CGED seeks to identify grammatical errors from sentences written by non-native learners of Chinese (Yu et al., 2014). It includes four kinds of errors, including insertion, replacement, redundant, and ordering. The dataset of CGED composes of sentence pairs, of which each sentence pair includes an erroneous sentence and an error-free sentence corrected by annotators. However, these sentence pairs do not provide information about erroneous positions, which are indispensable for the character replacement and character insertion. To obtain such position information, we implement a modified character alignment algorithm (Bryant et al., 2017) tailored for the Chinese language. Through this algorithm, we obtain a dataset for the insertion and replacement, both of which are suitable to examine the language learning ability of the pretrained model. We leave redundant and ordering types to future work. The statistic of our dataset is detailed in Appendix A.

3 Experiments

In this section, we first describe the BERT-style models that we examined, and then report numbers.

3.1 Chinese BERT Models

We describe the publicly available BERT models as well as the models we trained.

	Length = 1		Length = 2		Length > 3		Average	
Insertion	p@1	p@10	p@1	p@10	p@1	p@10	p@1	p@10
BERT-base	76.0	97.0	37.2	76.0	14.4	50.1	42.5	74.4
Ours-clm	**77.2**	**97.3**	36.7	74.4	13.3	49.3	42.4	73.7
Ours-wwm	56.6	80.1	**42.9**	79.1	19.3	**54.0**	39.6	71.1
Ours-clm-wwm	71.3	95.1	42.6	**80.9**	**20.6**	53.0	**44.8**	**76.3**
Replacememt	p@1	p@10	p@1	p@10	p@1	p@10	p@1	p@10
BERT-base	66.0	95.1	21.0	58.2	10.1	**46.1**	32.4	66.5
Ours-clm	**67.4**	**96.6**	20.4	58.3	7.4	36.9	31.7	63.9
Ours-wwm	34.8	68.2	25.7	65.3	7.4	35.2	22.6	56.2
Ours-clm-wwm	59.2	93.7	**26.5**	**66.4**	**12.4**	41.6	**32.7**	**67.2**

Table 1: Probing results on character replacement and insertion.

Character Replacement

Input	Label	Prediction
我没有权利破害别人的生活 *(En: I have no right to destroy other people's lives.)*	坏	坏 (99.97%)
代沟问题越来越深刻。 *(En: The problem of generation gap is getting worse.)*	严重	严 (79.94%) 重 (91.85%)

Character Insertion

Input	Label	Prediction
吸烟不但对自己的健康 好，而且对非吸烟者带来不好的影响。 *(En: Smoking is not only bad for your health, but also bad to non-smokers.)*	不	不 (99.98%)
我下次去北京的时候，一定要吃北京烤鸭，我们在北京吃过的是越南料理等外国的 。 *(En: Next time I go to Beijing, I can not miss the Peking Duck. What we have eaten in Beijing are Vietnamese cuisine and other foreign dishes.)*	饭菜	美 (40.66%) 食 (33.55%)

Figure 2: Top predictions of Ours-clm-wwm for replacement and insertion types. For each position, probability of the top prediction is given in parenthesis. The model makes the correct prediction for top three examples. For the bottom example, the prediction also makes sense, although it is different from the ground truth.

As mentioned earlier, BERT-base (Devlin et al., 2018)[5] is trained with the standard MLM objective.[6] To make a fair comparison of CLM and WWM, we train three simple Chinese BERT baselines from scratch[7]: (1) Ours-clm: we train this model using CLM. (2) Ours-wwm: this model only differs in that it is trained with WWM. (3) Ours-clm-wwm: this model is trained with both CLM and WWM objectives. We train these three models on a text corpus of 80B characters consisting of news, wiki, and novel texts. For the WWM task, we use a public word segmentation tool Texsmart (Zhang et al., 2020) to tokenize the raw data first. The mask rate is 15% which is commonly used in existing works. We use a max sequence length of 512, use the ADAM optimizer (Kingma and Ba, 2014) with a batch size of 8,192. We set the learning rate to 1e-4 with a linear optimizer with 5k warmup steps and 100k training steps in total. Models are trained on 64 Tesla V100 GPUs for about 7 days.

3.2 Probing Results

We present the results on two probing tasks here. Models are evaluated by Prediction @k, denoting whether the ground truth for each position is covered in the top-k predictions. From Table 1, we can make the following conclusions. First, Ours-clm consistently performs better than Ours-wwm on probing tasks that one character needs to be replaced or inserted. We suppose this is because WWM would lose the association between characters corresponding to a word. Second, WWM is crucial for better performance when there is more than one character that needs to be corrected. This phenomenon can be observed from the results of Ours-wwm and Ours-clm-wwm, which both adopt WWM and perform better than Ours-clm. Third, pretrained with a mixture of CLM and WWM, Ours-clm-wwm performs better than Ours-wwm in the one-character setting and does better than

[5] https://github.com/google-research/bert/blob/master/README.md

[6] We do not compare with RoBERTa-wwm-ext because the released version lacks of the language modeling head.

[7] We also further train these models initialized from RoBERTa and BERT and results are given in Appendix B.

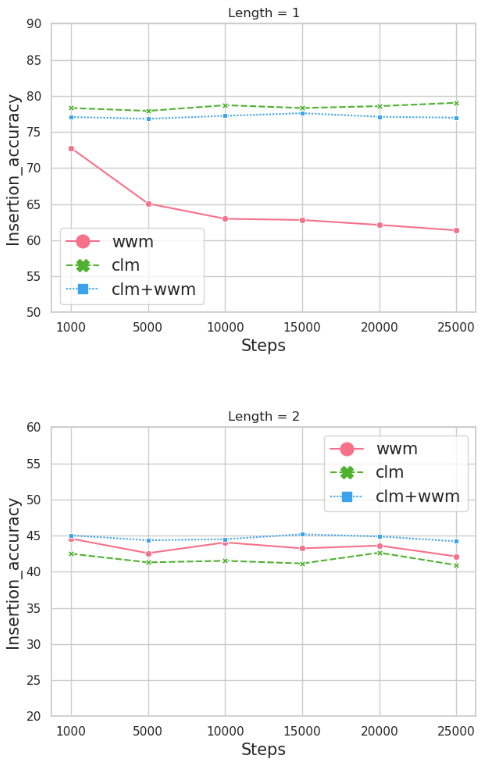

Figure 3: Model performance at different training steps on the probing task of character insertion. The top and bottom figures give the results evaluated on spans with one and two characters, respectively.

Ours-clm when more than one characters need to be handled. For each probing task, two examples with predictions produced by Ours-clm-wwm are given in Figure 2.

3.3 Analysis

To further analyze how CLM and WWM affect the performance on probing tasks, we initialized our model from RoBERTa (Cui et al., 2019) and further trained baseline models. We show the performance of these models with different training steps on the insertion task. From Figure 3 (top), we can observe that as the number of training steps increases, the performance of Ours-wwm decreases.

In addition, we also evaluate the performance of trained BERT models on downstream tasks with model parameters fine-tuned. The performance of Ours-clm-wwm is comparable with Ours-wwm and Ours-clm. More information can be found in Appendix C.

4 Related Work

We describe related studies on Chinese BERT model and probing of BERT, respectively.

The authors of BERT (Devlin et al., 2018) provided the first Chinese BERT model which was trained on Chinese Wikipedia data. On top of that, Cui et al. (2019) trained RoBERTa-wwm-ext with WWM on extended data. Cui et al. (2020) further trained a Chinese ELECTRA model and MacBERT, both of which did not have `[MASK]` tokens. ELECTRA was trained with a token-level binary classification task, which determined whether a token was the original one or artificially replaced. In MacBERT, `[MASK]` tokens were replaced with synonyms and the model was trained with WWM and ngram masking. ERNIE (Sun et al., 2019) was trained with entity masking, similar to WWM yet tokens corresponding to an entity were masked at once. Language features are considered in more recent works. For example, AMBERT (Zhang and Li, 2020) and Lattice-BERT (Lai et al., 2021) both take word information into consideration. ChineseBERT (Sun et al., 2021) utilizes pinyin and glyph of characters.

Probing aims to examine the language understanding ability of pretrained models like BERT when model parameters are clamped, i.e., without being fine-tuned on downstream tasks. Petroni et al. (2019) study how well pretrained models learn factual knowledge. The idea is to design a natural language template with a `[MASK]` token, such as "`the wife of Barack Obama is [MASK].`". If the model predicts the correct answer "`Micheal Obama`", it shows that pretrained models learn factual knowledge to some extent. Similarly, Davison et al. (2019) study how pretrained models learn commonsense knowledge and Talmor et al. (2020) examine on tasks that require symbolic understanding. Wang and Hu (2020) propose to probe Chinese BERT models in terms of linguistic and world knowledge.

5 Conclusion

In this work, we present two Chinese probing tasks, including character insertion and replacement. We provide three simple pretrained models dubbed Ours-clm, Ours-wwm, and Ours-clm-wwm, which are pretrained with CLM, WWM, and a combination of CLM and WWM, respectively. Ours-wwm is prone to lose the association between words and result in poor performance on probing tasks when one character needs to be inserted or replaced. Moreover, WWM plays a key role when two or more characters need to be corrected.

References

Christopher Bryant, Mariano Felice, and Edward Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.

Joe Davison, Joshua Feldman, and Alexander M Rush. 2019. Commonsense knowledge mining from pre-trained models. 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 1173–1178.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yuxuan Lai, Yijia Liu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2021. Lattice-bert: Leveraging multi-granularity representations in chinese pre-trained language models. *arXiv preprint arXiv:2104.07204*.

Lung-Hao Lee, Gaoqi Rao, Liang-Chih Yu, Endong Xun, Baolin Zhang, and Li-Ping Chang. 2016. Overview of NLP-TEA 2016 shared task for Chinese grammatical error diagnosis. In *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)*, pages 40–48, Osaka, Japan. The COLING 2016 Organizing Committee.

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*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Gaoqi Rao, Qi Gong, Baolin Zhang, and Endong Xun. 2018. Overview of NLPTEA-2018 share task Chinese grammatical error diagnosis. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 42–51, Melbourne, Australia. Association for Computational Linguistics.

Gaoqi Rao, Erhong Yang, and Baolin Zhang. 2020a. Overview of nlptea-2020 shared task for chinese grammatical error diagnosis. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 25–35.

Gaoqi Rao, Erhong Yang, and Baolin Zhang. 2020b. Overview of NLPTEA-2020 shared task for Chinese grammatical error diagnosis. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 25–35, Suzhou, China. Association for Computational Linguistics.

Gaoqi Rao, Baolin Zhang, Endong Xun, and Lung-Hao Lee. 2017. IJCNLP-2017 task 1: Chinese grammatical error diagnosis. In *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 1–8, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, Xiang Ao, Qing He, Fei Wu, and Jiwei Li. 2021. Chinesebert: Chinese pretraining enhanced by glyph and pinyin information. *arXiv preprint arXiv:2106.16038*.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. olmpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

Zhiruo Wang and Renfen Hu. 2020. Intrinsic knowledge evaluation on chinese language models. *arXiv preprint arXiv:2011.14277*.

C. Wood and V. Connelly. 2009. Contemporary perspectives on reading and spelling.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. 2020a. CLUE: A Chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, et al. 2020b. Clue: A chinese language understanding evaluation benchmark. *arXiv preprint arXiv:2004.05986*.

Liang-Chih Yu, Lung-Hao Lee, and Liping Chang. 2014. Overview of grammatical error diagnosis for learning chinese as a foreign language. In *Proceedings of the 1stWorkshop on Natural Language Processing Techniques for Educational Applications (NLP-TEA'14)*, pages 42–47.

Haisong Zhang, Lemao Liu, Haiyun Jiang, Yangming Li, Enbo Zhao, Kun Xu, Linfeng Song, Suncong Zheng, Botong Zhou, Jianchen Zhu, Xiao Feng, Tao Chen, Tao Yang, Dong Yu, Feng Zhang, Zhanhui Kang, and Shuming Shi. 2020. Texsmart: A text understanding system for fine-grained ner and enhanced semantic analysis. *arXiv preprint arXiv:2012.15639*.

Xinsong Zhang and Hang Li. 2020. Ambert: A pre-trained language model with multi-grained tokenization. *arXiv preprint arXiv:2008.11869*.

A The statistic of dataset

	Replacement	Insertion	Total
Length = 1	5,522	4,555	10,077
Length = 2	2,004	1,337	3,341
Length≥ 3	305	383	688
No. sentences	5,727	4,721	10,448
No. spans	7,831	6,275	14,106
No. chars	10,542	8,533	19,075

Table 2: The statistic of our dataset.

B Probing results from models with different initialization

We also verify the performance of models initialized from BERT (Devlin et al., 2018) and RoBERTa (Cui et al., 2019) on probing tasks. The results are detailed in Table 3, from which we can obtain consistent conclusions with the previous section.

C The evaluation on downstream tasks

We test the performance of BERT-style models on tasks including text classification (TNEWS, IFLYTEK), sentence-pair semantic similarity (AFQMC), coreference resolution (WSC), key word recognition (CSL), and natural language inference (OCNLI) (Xu et al., 2020a). We follow the standard fine-tuning hyper-parameters used in Devlin et al. (2018); Xu et al. (2020b); Lai et al. (2021) and report results on the development sets. The detailed results is shown in Table 4.

	Initialization	Length = 1		Length = 2		Length > 3		Average	
Insertion		p@1	p@10	p@1	p@10	p@1	p@10	p@1	p@10
BERT-base		76.0	97.0	37.2	76.0	14.4	50.1	42.5	74.4
Ours-clm	from scratch	77.2	97.3	36.7	74.4	13.3	49.3	42.4	73.7
Ours-wwm		56.6	80.1	42.9	79.1	19.3	54.0	39.6	71.1
Ours-clm-wwm		71.3	95.1	42.6	80.9	20.6	53.0	**44.8**	**76.3**
Ours-clm	from BERT	79.2	97.7	40.0	77.6	16.2	53.5	45.1	76.3
Ours-wwm		61.2	87.7	43.4	79.4	20.1	56.4	41.6	74.5
Ours-clm-wwm		73.1	96.1	41.8	80.6	20.6	56.7	**45.2**	**77.8**
Ours-clm	from RoBERTa	79.4	97.9	42.0	80.4	20.6	52.3	47.3	76.9
Ours-wwm		61.4	87.9	44.3	79.9	20.1	59.3	41.9	75.7
Ours-clm-wwm		77.3	97.5	46.8	83.3	22.5	58.7	**48.9**	**79.8**
Replacememt		p@1	p@10	p@1	p@10	p@1	p@10	p@1	p@10
BERT-base		66.0	95.1	21.0	58.2	10.1	46.1	32.4	66.5
Ours-clm	from scratch	67.4	96.6	20.4	58.3	7.4	36.9	31.7	63.9
Ours-wwm		34.8	68.2	25.7	65.3	7.4	35.2	22.6	56.2
Ours-clm-wwm		59.2	93.7	26.5	66.4	12.4	41.6	**32.7**	**67.2**
Ours-clm	from BERT	69.0	96.9	24.5	64.7	8.4	47.3	**34.0**	69.6
Ours-wwm		40.6	81.6	27.2	67.9	8.4	39.4	25.4	63.0
Ours-clm-wwm		61.6	94.9	27.6	67.8	10.4	47.0	33.2	**69.9**
Ours-clm	from RoBERTa	69.7	96.8	26.7	68	12.1	51.7	36.2	72.2
Ours-wwm		41.7	80.9	28.2	68.2	12.4	47.2	27.4	65.4
Ours-clm-wwm		67.3	96.7	28.4	69.7	15.7	54.2	**37.1**	**73.5**

Table 3: Probing results from models with different initialization.

Model		TNEWS	IFLYTEK	AFQMC	OCNLI	WSC	CSL	Average
BERT-base		57.1	61.4	74.2	75.2	78.6	81.8	71.4
Ours-clm	from scratch	57.3	60.3	72.8	73.9	79.3	68.7	68.7
Ours-wwm		57.6	60.9	73.8	75.4	81.9	75.4	**70.8**
Ours-clm-wwm		57.3	60.3	72.3	75.6	79.0	79.5	70.7
Ours-clm	from BERT	57.6	60.6	72.8	75.5	79.3	80.1	71.0
Ours-wwm		58.3	60.8	71.73	76.1	79.9	80.7	**71.3**
Ours-clm-wwm		58.1	60.8	72.3	75.8	80.3	79.9	71.2
Ours-clm	from RoBERTa	57.9	60.8	74.7	75.7	83.1	82.1	72.4
Ours-wwm		58.1	61.1	73.9	76.0	82.6	81.7	72.2
Ours-clm-wwm		58.1	61.0	74.0	75.9	84.0	81.8	**72.5**

Table 4: Evaluation results on the dev set of each downstream task. Model parameters are fine-tuned.