1Cademy at Semeval-2022 Task 1: Investigating the Effectiveness of Multilingual, Multitask, and Language-Agnostic Tricks for the Reverse Dictionary Task

Zhiyong Wang[1 4*], **Ge Zhang**[2 4*], **Nineli Lashkarashvili**[3 4]
[1]University of Colorado Boulder, USA
[2]University of Michigan Ann Arbor, USA
[3]San Diego State University, USA
[4]1Cademy Community, USA
Zhiyong.Wang@colorado.edu

Abstract

This paper describes our system for the SemEval2022 task of matching dictionary glosses to word embeddings. We focus on the Reverse Dictionary Track of the competition, which maps multilingual glosses to reconstructed vector representations. More specifically, models convert the input of sentences to three types of embeddings: SGNS, Char, and Electra. We propose several experiments for applying neural network cells, general multilingual and multitask structures, and language-agnostic tricks to the task. We also provide comparisons over different types of word embeddings and ablation studies to suggest helpful strategies. Our initial transformer-based model achieves relatively low performance. However, trials on different retokenization methodologies indicate improved performance. Our proposed Elmo-based monolingual model achieves the highest outcome, and its multitask, and multilingual varieties show competitive results as well.

1 Introduction

Reverse dictionary Task is defined as word generation based on user descriptions (Hill et al., 2016). Following competition rules, pre-trained models and external information should be avoided, and large-scale language models are unsuitable for the task. Our paper is devoted to the performance comparison of different neural network structures, multilingual and multitask tricks, and elaborating on language-agnostic or bidirectional structure helpfulness. The competition (Mickus et al., 2022) has significant potential in contributing pretraining process acceleration, low-resource language model development, and commonsense using. Furthermore, the task is of high importance for explainable AI and natural language processing since it models direct mapping from human-readable data to machine-readable data.

Known word representation methods using dictionaries, knowledge databases, or glosses have been a common approach for years. Related models can be divided into two major groups. In the former, category methods highly rely on large-scale model construction. Levine et al. (2019) develop SenseBert, introducing super-senses from Wordnet (Miller, 1995) into general Bert model. Ernie (Sun et al., 2019) combines node embeddings from knowledge graph and matched entities to enhance word representations. KnowBert (Peters et al., 2019) subsumes the entity connection and Bert models, which are trained together. There are similar research works relevant to the topic (Wang et al., 2021, 2020; Yin et al., 2020). Still, their models' performances are dependent on the basic large-scale language model trained by sentence samples. In the latter group, traditional dependency-based language models learn directly from word dependency and glosses. They have two major disadvantages: incompatibility with modern language models and relatively low performance (Tissier et al., 2017; Levy and Goldberg, 2014; Wieting et al., 2015). There is ambiguity about whether recent embeddings and dictionary glosses are mappable from each other.

The paper specifically focuses on progressing utilization of the glosses, different word representations, and languages. **First**, we discuss ablation studies for language-agnostic trick, bidirectional, multilingual, and multitask models and explain the experimental results. **Second**, we apply and analyze different re-tokenization methods. **Finally**, we give instructive conclusions about encoder structures, distinctive word representation relations, and cross-lingual dictionary performance based on our experiment results. We find that **(1)** transformer-based model performance is inferior to other models for its high complexity, **(2)** bidirectional models with similar parameter size outperform the unidirec-

[*] The two authors contributed equally to this work.

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 15 - 22

tional model because of their better understanding of context-environments even in the low-resource condition, and **(3)** different word embeddings have a potential relations and can be collaboratively learnt from glosses using a multitask learning structure. We make our codes and results publicly available[1].

2 Task Description

The competition, comparing dictionaries and word embeddings, proposes definition modeling (Noraset et al., 2017) and reverse dictionary sub-tracks (Hill et al., 2016). These sub-tracks are designed to test the equivalency of dictionary glosses and word embedding representations. This paper focuses on the reverse dictionary direction. The task refers to word recalling using gloss input and provides word representations that are separately generated by word2vec (SGNS) (Mikolov et al., 2013), character (Wieting et al., 2016), and Electra (Clark et al., 2020) embeddings as training data. External data and large-scale language models are strictly restricted from this competition since the models might learn the word embeddings majorly from the sentence samples instead of the dictionary glosses. The words matched with the dictionary glosses are hidden in the datasets, implying that dependency-based word representation algorithms cannot be applied directly.

3 Methodology

To clarify, we affirm that we only refer to the model structures instead of the trained models when we mention Elmo and MBert in the section and use no external data.

3.1 Language Model Structure

Baseline monolingual models with five distinctive structures were trained: RNN, LSTM, Bi-RNN, Elmo, and Transformer.

We experiment how bidirectional and different feature generator cell structures help.

RNN is the classical deep learning model dealing with ordinal or sequential data (Zaremba et al., 2014). Its major disadvantage is the vanishing and exploding gradient issue. Nevertheless, the model is fast to converge and works well on smaller sentences. Our experiments show that RNN, having similar results to the LSTM-based model, performs slightly better than the transformer-based one.

[1] https://github.com/ravenouse/Revdict_1Cademy

LSTM is another classical and widely-used feature generator structure in natural language processing. The comparison of LSTM-based and RNN-based models can suggest whether vector representation of glosses suffers from the long-term dependencies problem. Earlier works (Jozefowicz et al., 2015) demonstrate that variants of LSTM achieve similar performances in the majority of natural language processing tasks. We select the classical LSTM structure for the experiments.

Transformer (Vaswani et al., 2017) is a milestone feature extractor allowing deeper neural network design for natural language processing tasks. However, given the much smaller size of the competition data, it performs relatively worse compared to the expectation.

3.2 Multitask Structure

Although character embedding generation has a similar algorithm to general word embedding methods, it focuses on character representation and is mightier to better tackle the Out Of Vocabulary (**OOV**) problem. We applied Mean Squared Loss (**MSE Loss**) and Dynamic Weight Averaging (**DWA**) (Liu et al., 2019) as a basic multitask structure for predicting word2vec, Char, and Electra embedding together. It achieves competitive performance in both tasks.

DWA (Liu et al., 2019) is designed for keeping different tasks converging at the same pace. N denotes the number of tasks, T adjusts the weight-changing sensitivity according to loss difference of the tasks, $L_n(t-1)$ and $r_n(t-1)$ represent the loss and the training speed of task n at $(t-1)$th step. $w_i(t)$ is the loss weight of task i at tth step. The key update equations can be expressed as follows:

$$w_i(t) = \frac{N\, exp(r_i(t-1)/T)}{\sum_n exp(r_i(t-1)/T)} \tag{1}$$

$$r_n(t-1) = \frac{L_n(t-1)}{L_n(t-2)} \tag{2}$$

3.3 Retokenize Algorithm

We tried 3 widely-used retokenization algorithms for vocabulary generation including Byte Pair Encoding (**BPE**) (Sennrich et al., 2015), **WordPiece** Model (Schuster and Nakajima, 2012), and Unigram Language Model (**ULM**) (Kudo, 2018). BPE is a greedy algorithm that can not model word relation probability successfully. WordPiece considers

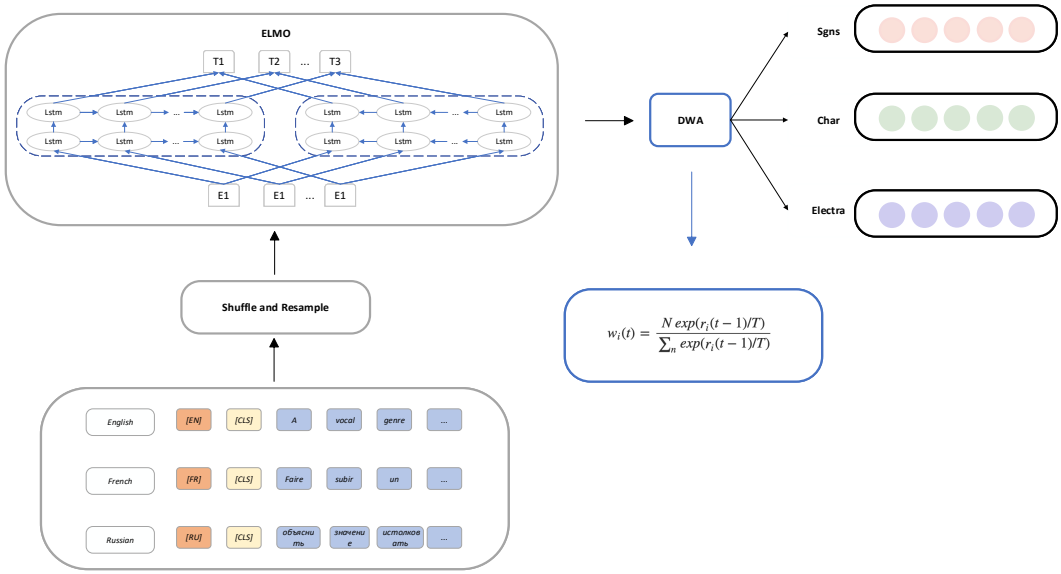

Figure 1: Sketch Map of the multilingual and multitask Elmo-based model structure.

word co-occurrence probability and is influenced by the source data. ULM assumes that all subwords are independent and the probability of a subword sequence is the multiple of its element subwords' probability.

3.4 Multilingual Structure

We applied two basic multilingual structures for the task: mBert (Pires et al., 2019) and adding the language tag. **MBert** has a shared vocabulary for all source languages. The results show that mBert can successfully model similar grammar structure, and sentences with similar meanings have akin representations using mBert. By applying mBert structure, we can estimate how these important conclusions would work for the reverse vocabularies task. We **add the language tag** as the first token to improve models' ability to separate different languages' representations.

We speculate that language-agnostic representations might aid multilingual models in achieving better performance. Residual connection cutting trick proposed by (Liu et al., 2020) was tried, to test how the research findings would work for our specific task.

3.5 Selected Model Design

Following experiment results and ablation studies, our best model is the monolingual Elmo with WordPiece tokenizer. The Multitask and multilingual tricks have proved to achieve competitive results with the Elmo language model. Adding language tokens achieves a better performance than the plain mBert structure while the Residual Cutting trick does not. It implies that the language-specific information is beneficial for the multilingual word representations of the reverse dictionaries task. Adding language tokens has demonstrated to help the Elmo-based multilingual model as well. The most promising multilingual and multitask Elmo-based model structure is shown in Figure 1.

4 Results and Discussion

4.1 Implementation Details

We apply **Bidirectional RNN** and **Elmo** (Peters et al., 2018) models with the same parameter size to find whether bidirectional structure helps. We selected AdamW (Loshchilov and Hutter, 2017) as optimizer. All monolingual models share the same hyper-parameters: the number of layers - 4, the hidden/input size - 256, and the dropout rate - 0.3. WordPiece tokenization was used as the best model design. We follow Devlin et al. (2019) to set the [CLS] token as the first token for monolingual models. We keep the [CLS] token when adding language tokens but set the language token as the first token instead.

Word Representations	SGNS			Char			Electra		
Monolingual Models	MSE	COS	RANK	MSE	COS	RANK	MSE	COS	RANK
RNN+WordPiece	1.000	0.249	0.310	0.158	0.778	0.442	**1.454**	**0.832**	0.433
LSTM+WordPiece	0.990	0.228	0.375	0.148	0.791	0.458	1.491	0.831	0.449
Transformer+WordPiece	1.042	0.214	0.367	0.194	0.780	0.453	1.796	0.827	0.486
BiRNN+WordPiece	0.989	0.221	0.395	0.150	0.791	0.454	1.483	0.832	0.449
Elmo+WordPiece	1.041	0.252	0.282	0.161	0.772	0.430	1.512	0.829	0.434
Elmo+BPE	1.037	0.250	**0.250**	0.162	0.774	0.443	1.537	0.822	0.436
Elmo+ULM	1.022	**0.265**	0.259	0.157	0.781	**0.430**	1.525	0.829	0.432
Elmo+WordPiece+DWA	**0.985**	0.246	0.298	**0.142**	**0.799**	0.447	1.514	0.827	**0.428**

Table 1: Experiment results on English resource test data using the monolingual models. Check section 2 for word algorithm representations' abbreviation. Check section 3 for details of monolingual models.

4.2 Main Results

Reverse dictionary results are evaluated using three metrics: mean squared error (**MSE**) between the reconstructed and reference embeddings, cosine similarity (**COS**) between the reconstructed embedding and the reference embedding, and the cosine-based ranking (**RANK**) between the reconstructed and reference embeddings, measuring the number of other test items having higher cosine with the reconstructed embedding than with the reference embedding(Mickus et al., 2022).

4.2.1 Monolingual Model Performance

We show monolingual models' results in Table 1. As depicted, our proposed model demonstrates competitive if not the best results across the metrics. English, for having the most detailed dictionary data, is selected to present monolingual models' performance[2].

We notice that the transformer-based model has inferior performance on the task. The competition provides a low-resource data set that can explain poorer outcomes for models with high complexity. We tried unidirectional and bidirectional models with similar feature extractors and parameter sizes. The results confirm that bidirectional models perform better and benefit from grasping the context-environment more accurately.

4.2.2 Multilingual Model Performance

We show two ablation experiment results to explain the influence of adding language tags and residual connection removal. **First**, experiment results of the Transformer-based multilingual model on SGNS embedding can suggest the benefits of language tags and curbing residual connection separately or jointly. **Second**, we propose experimental results of the original and adjusted Elmo-based multilingual models. The latter subsumes added language tokens. Such a comparison would clarify whether adding language tokens lead to a general improvement across different languages and word representations.

Electra word representations of Spanish and Italian are not available, implying no related experimental results. The outcomes demonstrate that multilingual models benefit from language-specific information but not from language-agnostic structure. Adding language tags has proved a positive influence on various language models.

4.3 Ablation Study

4.3.1 Tokenizer

We tried three widely-used tokenizers for our proposed model: BPE, ULM, and WordPiece. Both ULM and WordPiece show competitive performance in transformer- and Elmo-based structures. BPE has relatively low performance since the data resource is insufficient and has higher resource requests.

4.3.2 Multitask Model

According to the performance comparison in Table 1, DWA helps the Elmo model achieve better performance and reconstructs three-word representations simultaneously. It demonstrates that differently learned word representations have an internal relation and can be learned together using a shared bottom structure.

[2]check Table 5

Languages	EN			ES			FR			IT			RU		
Multilingual Models	MSE	COS	RANK	MSE	COS	RANK	MSE	COS	RANK	MSE	COS	RANK	MSE	COS	RANK
Transformer	1.023	0.201	0.400	0.977	0.300	**0.310**	1.051	0.278	**0.338**	1.143	**0.280**	**0.340**	0.564	0.318	0.363
Transformer+RC	1.029	0.199	0.417	1.005	0.298	0.329	1.069	0.253	0.374	1.189	0.267	0.364	0.601	0.279	0.409
Transformer+ALT	1.043	**0.215**	**0.397**	1.014	**0.308**	0.310	1.103	**0.280**	0.350	1.158	0.276	0.341	0.603	**0.326**	**0.337**
Transformer+RC+ALT	**1.011**	0.159	0.500	**0.955**	0.266	0.422	**1.044**	0.271	0.360	**1.129**	0.264	0.376	**0.561**	0.308	0.371

Table 2: Experiment results on **SGNS** word representation using the multilingual Transformer-based models. Check section 3 for details of multilingual models. **RC** represents the Residual Cutting trick. **ALT** represents the Adding Language Token trick.

Word Representations	SGNS			Char			Electra		
Multilingual Models	MSE	COS	RANK	MSE	COS	RANK	MSE	COS	RANK
Elmo_EN	1.023	0.238	0.317	0.177	0.759	**0.447**	1.555	0.818	**0.440**
Elmo+ALT_EN	**1.014**	**0.246**	**0.300**	**0.164**	**0.762**	0.449	**1.540**	**0.825**	0.441
Elmo_ES	**0.953**	0.342	**0.234**	0.532	0.810	0.405	NA	NA	NA
Elmo+ALT_ES	0.960	**0.351**	0.235	**0.511**	**0.822**	**0.393**	NA	NA	NA
Elmo_IT	**1.094**	0.343	**0.218**	0.355	0.720	0.403	NA	NA	NA
Elmo+ALT_IT	1.106	**0.343**	0.214	**0.354**	**0.735**	**0.387**	NA	NA	NA
Elmo_FR	**1.001**	0.313	0.255	0.388	0.752	**0.411**	1.298	0.845	0.445
Elmo+ALT_FR	1.004	**0.321**	**0.246**	**0.387**	**0.757**	0.411	**1.228**	**0.859**	**0.439**
Elmo_RU	**0.547**	0.357	0.247	0.145	0.816	**0.398**	0.891	**0.729**	0.386
ELmo+ALT_RU	0.563	**0.368**	**0.232**	**0.137**	**0.828**	0.400	**0.887**	0.728	**0.384**

Table 3: Experiment results of the multilingual ELmo-based models. **ALT** represents the Adding Language Token trick.

4.3.3 Difficulty of Reconstructing Different Word Representations

Compared with the Char and Electra, we find that the SGNS is harder to learn from the gloss corpus, suggesting that the contextualized information of words in sentences might be missing from the pure dictionary glosses. Additionally, the result along with (Kaneko and Bollegala, 2021) indicates dictionary corpus can be a promising way to remove the unfair biases rooted in large corpus learned word embeddings.

4.3.4 Difficulty of Learning Different Languages

Languages	Gloss Num	Dict.Size	Avg.Gloss Len	Elmo SGNS COS
English	43608	29042	11.7	0.252
French	43608	40028	14.3	0.333
Italian	43608	40126	13.6	0.352
Spanish	43608	46761	14.8	0.362
Russia	43608	57137	11.3	0.387

Table 4: Language Vocabulary Size Ablation Study. **Dict. Size** means the number of non-repeating tokens shown in the glosses. **Avg. Gloss Len** means the average token numbers contained in a gloss.

Our results of experiments show a strong positive correlation between language's tokens dictionary size and the models' achievable performance Table 4.

There are several possible reasons for the observation. First, as the language model dictionary size decreases, the models' and glosses' ability to explain the slight differences between words, especially the polysemies and synonyms, decreases. Second, a smaller dictionary size indicates that the covered tokens in the language model are a relatively incomplete part of words of the language.

Noted that the second explanation above does not consider the intrinsic differences between languages. The morphologically rich languages, like Russian, tend to have larger vocabulary sizes and bring many unknown words that influence performance negatively (Jurafsky and Martin, 2020).

5 Conclusion

The paper proposes a model showing competitive results in most cases of the reverse dictionaries task. Several conclusions are provided about the reverse

dictionaries task by the paper based on the ablation studies. **First**, the transformer-based model, for its high complexity, performs worse compared to RNN- or LSTM-based models. Multilingual transformer-based model benefits from specifying languages and including language-related grammar positional information. **Second**, bidirectional models with similar parameter sizes outperform the unidirectional one since they better grasp the context in low-resource conditions. **Third**, different word representations are potential connections and can be collaboratively learned from glosses using a multitask learning structure. SGNS embedding is much harder to model compared to Character embedding and Electra embedding.

6 Acknowledgements

We express our gratitude to Prof. Shi Wang and Prof. Alexis Palmer for providing computing resources and guidance. We are grateful to the organizers for providing such a fascinating and inspiring competition and promptly resolving all our questions. Special thanks to Rebecca Lee, Xingran Chen, and Natalia Wojarnik for idea sharing and a deep discussion in the initial stage.

References

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

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.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30.

Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. 2015. An empirical exploration of recurrent network architectures. In *International conference on machine learning*, pages 2342–2350. PMLR.

Dan Jurafsky and James H Martin. 2020. *Speech and language processing*, 3rd edition draft edition.

Masahiro Kaneko and Danushka Bollegala. 2021. Dictionary-based debiasing of pre-trained word embeddings. *arXiv preprint arXiv:2101.09525*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*.

Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2019. Sensebert: Driving some sense into bert. *arXiv preprint arXiv:1908.05646*.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308.

Danni Liu, Jan Niehues, James Cross, Francisco Guzmán, and Xian Li. 2020. Improving zero-shot translation by disentangling positional information. *arXiv preprint arXiv:2012.15127*.

Shikun Liu, Edward Johns, and Andrew J Davison. 2019. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1871–1880.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Timothee Mickus, Denis Paperno, Mathieu Constant, and Kees van Deemter. 2022. SemEval-2022 Task 1: Codwoe – comparing dictionaries and word embeddings. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Thirty-First AAAI Conference on Artificial Intelligence*.

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.

ME Peters, M Neumann, M Iyyer, M Gardner, C Clark, K Lee, and L Zettlemoyer. 2018. Deep contextualized word representations. arxiv 2018. *arXiv preprint arXiv:1802.05365*, 12.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5149–5152. IEEE.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

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*.

Julien Tissier, Christophe Gravier, and Amaury Habrard. 2017. Dict2vec: Learning word embeddings using lexical dictionaries. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 254–263.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Guihong Cao, Daxin Jiang, Ming Zhou, et al. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*.

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.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics*, 3:345–358.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Charagram: Embedding words and sentences via character n-grams. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1515, Austin, Texas. Association for Computational Linguistics.

Da Yin, Tao Meng, and Kai-Wei Chang. 2020. SentiBERT: A transferable transformer-based architecture for compositional sentiment semantics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3695–3706, Online. Association for Computational Linguistics.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

A Appendix: A

Check Table 5 for experiment results of the monolingual models.

B Appendix: B

Check Table 6 for selected multilingual models' performance.

Word Representations	SGNS			Char			Electra		
Monolingual Models	MSE	COS	RANK	MSE	COS	RANK	MSE	COS	RANK
	Language English								
RNN+WordPiece	1.000	0.249	0.310	0.158	0.778	0.442	**1.454**	**0.832**	**0.433**
LSTM+WordPiece	**0.990**	0.228	0.375	**0.148**	**0.791**	0.458	1.491	0.831	0.449
Transformer+WordPiece	1.042	0.214	0.367	0.194	0.780	0.453	1.796	0.827	0.486
BiRNN+WordPiece	0.989	0.221	0.395	0.150	0.791	0.454	1.483	0.832	0.449
Elmo+WordPiece	1.041	**0.252**	**0.282**	0.161	0.772	**0.430**	1.512	0.829	0.434
	Language Spanish								
RNN+WordPiece	0.936	0.358	0.225	0.512	0.822	0.402	NA	NA	NA
LSTM+WordPiece	**0.928**	0.334	0.287	**0.497**	**0.829**	0.418	NA	NA	NA
Transformer+WordPiece	1.011	0.307	0.313	0.577	0.828	0.432	NA	NA	NA
BiRNN+WordPiece	0.939	0.315	0.329	0.511	0.826	0.423	NA	NA	NA
Elmo+WordPiece	0.968	**0.362**	**0.207**	0.520	0.820	**0.396**	NA	NA	NA
	Language French								
RNN+WordPiece	0.975	0.329	0.254	0.379	0.761	0.408	1.272	0.856	0.444
LSTM+WordPiece	**0.971**	0.303	0.329	**0.361**	**0.772**	0.420	**0.191**	0.862	0.457
Transformer+WordPiece	1.057	0.273	0.366	0.461	0.771	0.430	1.523	0.856	0.488
BiRNN+WordPiece	0.984	0.290	0.361	0.366	0.770	0.424	1.202	**0.863**	0.454
Elmo+WordPiece	1.007	**0.333**	**0.239**	0.373	0.763	**0.402**	1.341	0.850	**0.437**
	Language Italian								
RNN+WordPiece	1.078	**0.353**	0.218	0.345	0.741	0.391	NA	NA	NA
LSTM+WordPiece	**1.077**	0.324	0.276	0.340	0.744	0.413	NA	NA	NA
Transformer+WordPiece	1.160	0.256	0.373	0.377	0.731	0.419	NA	NA	NA
BiRNN+WordPiece	1.086	0.309	0.303	**0.338**	**0.747**	0.415	NA	NA	NA
Elmo+WordPiece	1.106	0.352	**0.200**	0.354	0.736	**0.384**	NA	NA	NA
	Language Russian								
RNN+WordPiece	**0.537**	**0.388**	0.226	0.132	0.832	0.391	0.899	0.727	0.372
LSTM+WordPiece	0.547	0.338	0.346	**0.131**	**0.834**	0.401	**0.885**	**0.728**	0.400
Transformer+WordPiece	0.565	0.315	0.377	0.156	0.827	0.411	1.071	0.707	0.473
BiRNN+WordPiece	0.551	0.321	0.397	0.135	0.831	0.403	0.919	0.727	0.410
Elmo+WordPiece	0.557	0.387	**0.217**	0.134	0.831	**0.390**	0.904	0.723	**0.362**

Table 5: **Appendix A**. Experiment results of the monolingual models. Check section 2 for word algorithm representations' abbreviation. Check section 3 for details of monolingual models.

Word Representations	SGNS			Char			Electra		
Monolingual Models	MSE	COS	RANK	MSE	COS	RANK	MSE	COS	RANK
	Language English								
Elmo+WordPiece	1.041	**0.252**	**0.282**	0.161	0.772	**0.430**	**1.512**	**0.829**	0.434
Elmo + WordPiece + DWA	**0.985**	0.246	0.298	**0.142**	**0.799**	0.447	1.514	0.827	**0.428**
	Language French								
Elmo+WordPiece	1.007	**0.333**	**0.239**	0.373	0.763	**0.402**	1.341	0.850	0.437
Elmo + WordPiece + DWA	**0.937**	0.327	0.243	**0.364**	**0.770**	0.406	**1.315**	**0.854**	**0.428**
	Language Russian								
Elmo+WordPiece	0.557	0.387	0.217	0.134	0.831	**0.390**	0.904	0.7226	**0.362**
Elmo + WordPiece + DWA	**0.534**	**0.388**	**0.189**	**0.127**	**0.838**	0.376	**0.908**	**0.7235**	0.364

Table 6: **Appendix B**. The table shows the selected multilingual models' performance. Check section 2 for word algorithm representations' abbreviation. Check section 3 for details of monolingual models.