JSI at SemEval-2022 Task 1: CODWOE - Reverse Dictionary: Monolingual, multilingual, and cross-lingual approaches

Hanh Thi Hong TRAN[1,2,3], **Matej MARTINC**[1], **Matthew PURVER**[4], **Senja POLLAK**[1]
[1]Jožef Stefan Institute, Slovenia
[2]Jozef Stefan International Postgraduate School, Slovenia
[3]University of La Rochelle, France
[4]Queen Mary University of London, UK

Abstract

The *reverse dictionary* is a sequence-to-vector task in which a gloss is provided as input, and the model is trained to output a semantically matching word vector. The reverse dictionary is useful in practical applications such as solving the tip-of-the-tongue problem, helping new language learners, etc. In this paper, we evaluate the Transformer-based model with the added LSTM layer for the task at hand in a monolingual, multilingual, and cross-lingual zero-shot setting. Experiments are conducted in five languages in the CODWOE dataset, namely English, French, Italian, Spanish, and Russian. Our work partially improves the current baseline of the CODWOE competition and offers insight into the feasibility of the cross-lingual methodology for the reverse dictionary task. The code is available at https://github.com/honghanhh/codwoe2021.

1 Introduction

The CODWOE 2021 shared task on dictionary glosses and word embedding representations, organized as part of the SemEval workshop, presented one of the first opportunities to systematically study and compare these semantic descriptions by two sub-tracks: model definition and reverse dictionary.

While definition modeling consists in using the vector representation of e.g. “giraffe” to produce the associated gloss, e.g. “a tall, long-necked, spotted ruminant of Africa”, the reverse dictionary is the mathematical inverse: reconstruct an embedding for the word “giraffe” from the corresponding gloss. In this paper, we dive into the reverse dictionary task modelling to learn the ability to infer word embeddings from dictionary resources.

A reverse dictionary is useful in real-world applications. First of all, it can effectively solve the tip-of-the-tongue problem (Brown and McNeill, 1966): the inability to retrieve a word from memory. People who suffer from this problem such as copywriters, novelists, researchers, students, etc. can quickly and easily find the words they need thanks to reverse dictionary. Furthermore, new language learners who grasp a limited number of words can also take advantage of the reverse dictionary to express correctly. Besides, it plays an important role in word selection for anomia patients (Benson, 1979), who can recognize and describe an object but fail to name it due to neurological disorder.

The contributions of this paper are as follows:

1. We evaluate the performance of the Transformer-based model with an additional LSTM, BiLSTM, and the combination of both additional layers on separate languages as well as the performance of a multilingual model trained on the concatenated corpus containing text for all five given languages.

2. We analyze the effectiveness of zero-shot learning by training the model on a particular language and apply it for prediction on the rest.

This paper is organised as follows: Section 2 presents the related works in reverse dictionary. Next, we introduce our methodology in Section 3, and the experimental details in Section 4. The results are discussed in Section 5, before we conclude and present future works in Section 6.

2 Related Work

The reverse dictionary systems tend to employ two distinct approaches. The first approach takes advantage of sentence matching (Bilac et al., 2004; Zock and Bilac, 2004; Méndez et al., 2013; Shaw et al., 2011) to return the words whose dictionary definitions are most similar to the corresponding gloss.

The second approach focuses on neural language models to encode the glosses into a vector representation and returns the words with the closest embeddings to the vector of the glosses (Hill et al.,

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 101 - 106
July 14-15, 2022 ©2022 Association for Computational Linguistics

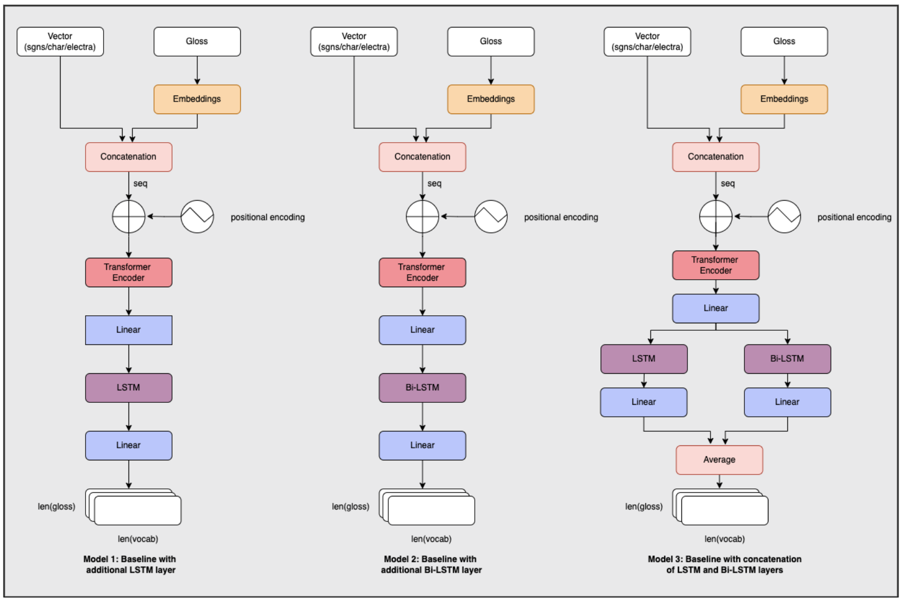

Figure 1: The overall model architecture.

2016; Kartsaklis et al., 2018; Morinaga and Yamaguchi, 2018; Hedderich et al., 2019; Pilehvar, 2019). As a result, the performance depends largely on the word representation's quality. However, many words are low-frequency and usually have poor embeddings regarding Zipf's law.

To tackle the above issue, a multi-channel reverse dictionary model has been proposed (Zheng et al., 2020; Qi et al., 2020). The system includes a sentence encoder (e.g. a BiLSTM (Hochreiter and Schmidhuber, 1997), BERT (Devlin et al., 2018)) with attention (Bahdanau et al., 2014), and diverse characteristic predictors that are useful to find the target words with poor representations and exclude wrong words with similar embeddings to the target words, for example, antonyms.

In terms of production, OneLook[1] and ReverseDictionary[2] are two successful commercial English reverse dictionary systems. However, their architectures are undisclosed and their performance is far from perfect. Meanwhile, open-sourced WantWords[3] (Qi et al., 2020) is a rising star with state-of-the-art (SOTA) performance in English and even competitive results in a cross-lingual Chinese-English and English-Chinese setting.

3 Methodology

As the competition does not allow the use of external data or pretrained language models in order to make approaches easily comparable, we start by experimenting with the simplest form of Transformer, a deep learning model that adopts the self-attention mechanism, differentially weighting the significance of each part of the input data. This is also the baseline shared by CODWOE's organizers. Then we experiment by adding an additional LSTM layer (Model 1), BiLSTM layer (Model 2), and combining the prediction from these two mentioned layers (Model 3). The overall architecture is presented in Figure 1.

The objective of the model is to map the glosses to the vector representation of the word that the gloss defines. The target embeddings are learned by a skip-gram with negative sampling (sgns) approach (word2vec). During training, the input is the gloss, which is tokenized using the Byte Pair Encoding (BPE) algorithm[4] and then converted into word embeddings. The positional encoding is applied to each embedding to inject meaningful information about the position of the tokens in the sequence. After that, they are fed into a Transformer Encoder, which is a stack of four identical encoder blocks. As illustrated in Figure 2, each block includes the following layers in the same order: a multi-head self-attention layer that explores the word correlations followed by a normalization layer (both of them are surrounded by a residual connection), and then a linear layer followed by a second normalization layer (both of them are also

[1] https://onelook.com/thesaurus/
[2] https://reversedictionary.org/
[3] https://wantwords.thunlp.org/
[4] We employ the SentencePiece library: https://github.com/google/sentencepiece

surrounded by a residual connection). A dropout layer is then added to avoid overfitting. In the baseline model suggested by the CODWOE's organizers, the results from the above architecture are then passed into a linear layer to achieve the final model.

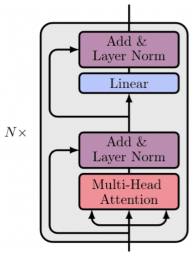

Figure 2: Transformer encoder (Vaswani et al., 2017).

We propose three settings regarding three different models constructed from the baseline architecture. We hypothesize that with an additional LSTM or BiLSTM layer, we can improve the modeling of the word-level sequential context, same as in (Wang et al., 2019), and therefore improve the performance of the model. In Model 1, we add one additional LSTM layer after the linear one. We take advantage of the BiLSTM layer in Model 2 to capture the information bidirectionally. We combine the result from the two mentioned layers by averaging their weights in Model 3. In the final step, we fed the LSTM or BiLSTM outputs into a linear layer to obtain the final vector representation. During the prediction phase, for each new data example, we feed the gloss into the trained model to obtain the vector presentation similar to the sgns.

The proposed three models are first tested in a monolingual setting, to determine which architecture achieves the best performance. Next, we explore if the target sgns embedding spaces may already be aligned to some degree across languages, even though the CODWOE organizers did not explicitly mention any cross-lingual alignment in the shared task description. We first attempt a multilingual experiment to examine the degree to which training in multiple languages affects performance. Finally, the best performing monolingual models are tested in a zero-shot cross-lingual setting, where we train the model in a specific language and evaluate it in different languages that the model has never seen before. The implementation details are in Section 4.2.

4 Experimental Setup

4.1 Dataset

The experiments were conducted on the dataset from the CODWOE 2021 competition. The data consists of glosses for five languages (English - **en**, Spanish - **es**, French - **fr**, Italian - **it**, and Russian - **ru** and three different word embedding representations for each gloss. In this paper, we focus only on skip-gram with negative sampling (sgns) embeddings trained on around 1 billion sentences in total with 50% of the sentences coming from Wikipedia, 40% coming from open subtitles, and the rest drawn from the corpora (e.g. Wikisource, gutenberg.org). All sentences were tokenized with the default NLTK's[5] tokenizer.

Each language contains 3 different sets, including the training set with 43,608 samples, the development set with 6,375 samples, and a test set containing 6,208 samples. Although the number of samples for each set is distributed equally among languages, a word can have a different number of glosses (polysemy), and vice versa, a gloss can belong to more than one word (synonymy).

Note that the training and development data hide the exact words matching each gloss and only release their sngs, char, and electra embeddings. However, on the full test set, the words are provided.

4.2 Experimental Settings

Due to time limitations, we have not conducted any hyperparameter search on the development sets over the space of possible model configurations, such as embedding dimension, learning rate, weight decay, size of hidden layers, etc. Alternatively, we decided to use a standard configuration based on previous research as well as suggested by the competition organizers for all the experiments. The configuration is presented in Table 1.

All models were implemented with Pytorch and trained on GPUs from Google Colab[6]. Further tuning and optimization will be left for future work.

4.3 Evaluation Metrics

The performance of the reverse dictionary system is evaluated by Mean squared error (MSE), Cosine similarity, and Cosine-based ranking (Dinu and Ionescu, 2012). These are the evaluation metrics suggested in the CODWOE 2021 competition,

[5] `https://www.nltk.org/`

[6] `https://colab.research.google.com/`

Table 1: Model configuration.

Settings	Values
Number of heads	4
Number of encoder layers	4
Number of epoches	20
Learning rate	1e-4
Weight decay	1e-6
Drop out	0.3
Optimizer	AdamW
Max length	512
Patience	5

which hereby facilitates the comparison between our approaches and the baseline. Further details about each evaluation metric can be found on the CODWOE 2021 website. Here, in this research, we aim to minimize the MSE and the cosine-based ranking, and maximize the cosine similarity.

5 Results

The test set results of our approach on the reverse dictionary task are presented in Table 2. We compare our three different models (LSTM, BiLSTM, and combined) with the baseline as well as with the winning approach on this shared task. In addition, we also present the results for a multilingual LSTM trained in all available languages.

In terms of MSE, the performance of the Transformer-based model with an additional LSTM layer is the most competitive for all languages except English when compared to our other approaches, namely BiLSTM and combined LSTM and BiLSTM. This model surpasses the baseline in Spanish and French according to most criteria. Meanwhile, the combination of the LSTM and BiLSTM layers after the Transformer encoder layer offers the best results on the English dataset, outperforming the baseline in terms of MSE. We also investigate a multilingual configuration where we train in all languages and employ the model on each language's test set. The results for the multilingual model are substantially lower compared to all other monolingual settings according to the MSE score. Compared to the best solution in the CODWOE competition proposed by WENGSYX team[7], the gap between our solution and theirs is on average 0.1 in terms of the MSE score.

Table 2: The evaluation results on the test dataset. We compare our models with additional LSTM, BiLSTM and combined LSTM and BiLSTM with the shared task baseline and the winning approach. We also test our multilingual approach trained on all languages of the train set. All the results above the baseline are in bold.

Language	Model	MSE	Cosine	Ranking
en	LSTM	0.913	**0.156**	0.499
en	BiLSTM	0.938	0.125	0.517
en	combined	**0.909**	0.139	0.513
en	multilingual LSTM	1.184	0.003	0.501
en	Baseline	0.911	0.151	0.490
en	#1 solution	**0.862**	**0.243**	**0.329**
es	LSTM	**0.914**	**0.223**	**0.499**
es	BiLSTM	1.031	0.005	**0.498**
es	combined	0.947	0.138	**0.495**
es	multilingual LSTM	0.978	**0.207**	**0.452**
es	Baseline	0.930	0.204	0.499
es	#1 solution	**0.858**	**0.353**	**0.251**
fr	LSTM	**1.123**	**0.216**	0.498
fr	BiLSTM	1.283	0.010	0.502
fr	combined	1.169	0.093	0.498
fr	multilingual LSTM	1.404	-0.005	0.524
fr	Baseline	1.140	0.198	0.491
fr	#1 solution	**1.030**	**0.328**	**0.282**
it	LSTM	1.201	-0.010	0.500
it	BiLSTM	1.287	-0.004	0.501
it	combined	1.208	-0.008	0.500
it	multilingual LSTM	1.305	-0.008	0.494
it	Baseline	1.125	0.204	0.477
it	#1 solution	**1.040**	**0.360**	**0.230**
ru	LSTM	0.616	0.006	0.500
ru	BiLSTM	0.795	-0.020	0.499
ru	combined	0.650	-0.016	0.499
ru	multilingual LSTM	0.934	-0.004	0.522
ru	Baseline	0.577	0.253	0.490
ru	#1 solution	**0.528**	**0.424**	**0.187**

In terms of Cosine similarity, the model with an additional LSTM layer proves to have better performance in English, Spanish, and French compared to other tested models. This model also surpasses the baseline model on Spanish and French test sets. In addition, the multilingual model also achieves a slightly better Cosine similarity than the baseline on the Spanish test set.

In terms of Cosine ranking, all models demonstrate a slightly higher ranking in comparison to the baseline on the Spanish test set, with the multilingual model achieving the best ranking. In other languages, the baseline model performs the best.

Overall, training the additional LSTM layer on a multilingual training set does not seem to improve the results compared to the monolingual settings, the only exception being the performance of the multilingual model on the Spanish test set in terms of Cosine ranking.

Given the fact that the Transformer-based model with an additional LSTM performs the best in a monolingual setting, we use this model for the zero-shot cross-lingual experiments. The results

[7] https://competitions.codalab.org/competitions/34022#results

Table 3: Cross-lingual zero-shot evaluation on test set.

Train set	Metrics	en	es	fr	it	ru
en	**MSE**	**0.913**	0.914	1.208	1.201	0.616
	Cosine	**0.156**	**0.223**	-0.020	-0.010	**0.006**
	Ranking	**0.499**	**0.499**	0.500	0.500	**0.500**
es	**MSE**	0.963	0.914	1.208	1.201	0.616
	Cosine	-0.004	**0.223**	-0.020	-0.010	**0.006**
	Ranking	0.501	**0.499**	0.500	0.500	**0.500**
fr	**MSE**	0.962	0.916	**1.123**	**1.198**	**0.615**
	Cosine	-0.004	0.215	**0.216**	**-0.005**	0.002
	Ranking	0.500	**0.499**	**0.498**	**0.499**	0.501
it	**MSE**	0.962	0.916	1.208	1.201	**0.615**
	Cosine	-0.004	0.215	-0.024	-0.010	0.002
	Ranking	0.501	**0.499**	0.501	0.500	0.501
ru	**MSE**	0.964	**0.913**	1.204	1.196	0.616
	Cosine	-0.004	0.222	-0.021	-0.010	**0.006**
	Ranking	0.501	0.500	0.500	0.500	**0.500**

for these experiments are displayed in Table 3. The first column indicates the language used for training and development, the second column displays the evaluation metrics including MSE, Cosine similarity, and Cosine ranking. The rest demonstrate the evaluation results of each metric on a specific test dataset per language. For example, in the first row where the training set is *en*, we train on the English training and development set and predict each of the five language's test sets.

In general, if the model is trained on a language matching the language of the test data, it performs better except in the French corpus. However, the interesting exception is that, for example, the Spanish test set, on which all models, no matter on which language they were trained, offer very consistent performance according to all measures. It is also interesting that the models trained in English and Spanish have exactly the same results on French, Italian, and Russian test sets. This might suggest that these models were not able to make sense of the examples in the test set and that their performance is on par with a random baseline. Further analysis of this behavior will be left for the future.

6 Conclusion

In this paper, we have investigated the performance of monolingual and multilingual Transformer-based models on the reverse dictionary problem, a sequence-to-vector task where a word representation needs to be constructed from the corresponding gloss. We have experimented with two additions to the original architecture, namely adding either an additional LSTM or a BiLSTM layer on top of the original architecture. We have also explored whether combining these two architectures improves the performance. Besides that, we explored the cross-lingual performance of the monolingual models and compared them to monolingual and multilingual classifiers.

On the task of reconstructing sgns embeddings, the monolingual Transformer-based model with an additional LSTM layer in most cases offers the best performance for English, Spanish, and French according to MSE and Cosine similarity. The model also offers competitive performance in terms of MSE for Italian and Russian compared to the baseline. Therefore, the results to some extent confirm the initial hypothesis that with an additional LSTM layer, we can improve the modeling of the word-level sequential context. Nevertheless, the improvements are worse than expected and the multilingual and zero-shot experiments yield unexpected results that require further analysis. We can therefore summarize our findings by saying that the reverse dictionary task of restoring sgns embeddings seems to be very challenging, and none of our models (and also other models in the competition) were able to successfully solve it, at least according to the scores achieved during the competition.

This means that there remains a lot of room for improvement. In the future, we would like to investigate the effect of different text representations on the performance of the model, e.g., by feeding the model graph representations. Combinations of several text representations will also be explored. Furthermore, the effectiveness of multilingual models compared to monolingual ones should be additionally explored. Despite zero-shot learning not working well in our studies, it is worth evaluating the performance of one-shot learning and few-shot learning with the hypothesis that the models can understand new concepts from only one or a few examples. Further experiments on the topic of adapting the Transformer architecture for the specific task at hand will also be conducted.

Acknowledgements

We acknowledge the Slovenian Research Agency ARRS for the core programme funding P2-103 "Knowledge Technologies". We also thank the region of La Nouvelle Aquitaine for funding the first author of this research.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

D Frank Benson. 1979. Neurologic correlates of anomia. In *Studies in neurolinguistics*, pages 293–328. Elsevier.

Slaven Bilac, Wataru Watanabe, Taiichi Hashimoto, Takenobu Tokunaga, and Hozumi Tanaka. 2004. Dictionary search based on the target word description. In *Proc. of the Tenth Annual Meeting of The Association for Natural Language Processing (NLP2004)*, pages 556–559.

Roger Brown and David McNeill. 1966. The "tip of the tongue" phenomenon. *Journal of verbal learning and verbal behavior*, 5(4):325–337.

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*.

Liviu P Dinu and Radu-Tudor Ionescu. 2012. A rank-based approach of cosine similarity with applications in automatic classification. In *2012 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 260–264. IEEE.

Michael A Hedderich, Andrew Yates, Dietrich Klakow, and Gerard De Melo. 2019. Using multi-sense vector embeddings for reverse dictionaries. *arXiv preprint arXiv:1904.01451*.

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.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Dimitri Kartsaklis, Mohammad Taher Pilehvar, and Nigel Collier. 2018. Mapping text to knowledge graph entities using multi-sense lstms. *arXiv preprint arXiv:1808.07724*.

Oscar Méndez, Hiram Calvo, and Marco A Moreno-Armendáriz. 2013. A reverse dictionary based on semantic analysis using wordnet. In *Mexican International Conference on Artificial Intelligence*, pages 275–285. Springer.

Yuya Morinaga and Kazunori Yamaguchi. 2018. Improvement of reverse dictionary by tuning word vectors and category inference. In *International Conference on Information and Software Technologies*, pages 533–545. Springer.

Mohammad Taher Pilehvar. 2019. On the importance of distinguishing word meaning representations: A case study on reverse dictionary mapping. 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 2151–2156.

Fanchao Qi, Lei Zhang, Yanhui Yang, Zhiyuan Liu, and Maosong Sun. 2020. Wantwords: An open-source online reverse dictionary system. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–181.

Ryan Shaw, Anindya Datta, Debra VanderMeer, and Kaushik Dutta. 2011. Building a scalable database-driven reverse dictionary. *IEEE Transactions on Knowledge and Data Engineering*, 25(3):528–540.

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.

Chenguang Wang, Mu Li, and Alexander J Smola. 2019. Language models with transformers. *arXiv preprint arXiv:1904.09408*.

Lei Zheng, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2020. Multi-channel reverse dictionary model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 312–319.

Michael Zock and Slaven Bilac. 2004. Word lookup on the basis of associations: from an idea to a roadmap. In *Proceedings of the Workshop on Enhancing and Using Electronic Dictionaries*, pages 29–35.