Edinburgh at SemEval-2022 Task 1: Jointly Fishing for Word Embeddings and Definitions

Pinzhen Chen **Zheng Zhao**
School of Informatics, University of Edinburgh
{pinzhen.chen, zheng.zhao}@ed.ac.uk

Abstract

This paper presents a winning submission to the SemEval 2022 Task 1 on two sub-tasks: reverse dictionary and definition modelling. We leverage a recently proposed unified model with multi-task training. It utilizes data symmetrically and learns to tackle both tracks concurrently. Analysis shows that our system performs consistently on diverse languages, and works the best with *sgns* embeddings. Yet, *char* and *electra* carry intriguing properties. The two tracks' best results are always in differing subsets grouped by linguistic annotations. In this task, the quality of definition generation lags behind, and BLEU scores might be misleading.

1 Introduction

We describe the University of Edinburgh's participation in SemEval 2022 Task 1 on comparing dictionaries and word embeddings (CODWOE), organized by Mickus et al. (2022).[1] The task features two directions: *reverse dictionary* and *definition modelling*. The former is to construct the embedding of a word given its definition gloss, and the latter is to generate the definition from a word embedding. The organizers provide datasets of word embedding-definition pairs across three types of embeddings and five languages. The training data has a size of 43.6k for each language, which is smaller than the data released in prior research (Hill et al., 2016; Chang et al., 2018). However, it provides a precious chance for a comprehensive study of lower-resourced reverse dictionary and definition modelling on languages other than English, as well as on different embedding architectures.

As our system architecture, we use a recently proposed unified model, which deals with both tracks concurrently and achieves superior results (Chen and Zhao, 2022). The model enables multi-task training by using word embeddings and definitions symmetrically. We also create ensembles and handcrafted phrases. Our code implementation builds on the organizers' and is publicly available.[2]

We submit to both reverse dictionary and definition modelling tracks, and cover all language and embedding combinations. Furthermore, we examine model generations and scores from three aspects: embedding architectures, languages, and linguistic annotations, aiming to figure out how these affect performance, subject to the models we have adopted. We finally show the information captured by different word embeddings and discuss the limitations in task evaluation and ranking.

Regarding the shared task outcome, we are the team with the most "gold medals": out of 18 sub-tracks, we attain first place in 8, second place in 4 and third place in 4. Our final ranks in the sub-tracks are detailed in Table 1.

Langauge		en	es	fr	it	ru
Reverse dictionary	sgns	2	4	3	2	3
	char	3	1	1	1	1
	electra	1	n/a	1	n/a	1
Definition modelling		4	3	2	2	1

Table 1: Our ranks in each sub-track.

2 Background

2.1 Datasets

The organizers provide datasets for five languages: English (*en*), Spanish (*es*), French (*fr*), Italian (*it*), and Russian (*ru*). Also, they supply 256d word embeddings from three architectures:

- *sgns*: static (non-contextualized) embeddings learned using skip-gram with negative sampling (Mikolov et al., 2013);
- *char*: character-based embeddings from an autoencoder trained on the spelling of a word;
- *electra*: contextualized embeddings produced by a generator-discriminator model (Clark et al., 2020).

[1] https://competitions.codalab.org/competitions/34022

[2] https://github.com/PinzhenChen/UnifiedRevdicDefmod

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 75 - 81
July 14-15, 2022 ©2022 Association for Computational Linguistics

Despite that *electra* is not available for *es* and *it*, the data still covers 13 combination. All embedding architectures are trained on comparable corpora for all languages. Participants are not allowed to use any external resources, and words are provided as embeddings rather than actual words.

For each language, data is split into train, validation, test, and trial sets, at sizes 43.6k, 6,4k, 6.2k, and 0.2k. Human annotations are included in the trial split for analysis, but only word embeddings and definition glosses can be used for training. The snippet below exemplifies a single data instance with all possible fields. Training, validation, and test sets consist of only the bolded key-value pairs; all fields are found in the tiny trial set.

{"**id**":"en.trial.2",
"**sgns**": [2.08729, 0.26177, ...],
"**char**": [0.38789, 0.19716, ...],
"**electra**": [-1.47715, -0.47424, ...],
"**gloss**": "A mixture of other substances or things .",
"word": "cocktail",
"pos": "noun",
"example": "a cocktail of illegal drugs",
"type": "hypernym-based",
"counts": 4187,
"f_rnk": 13245,
"concrete": 1,
"polysemous": 0}

2.2 Evaluation metrics and ranking

Reverse dictionary is evaluated by three metrics:

- *MSE*: mean squared error between references and generated embeddings;
- *cosine*: cosine similarity between references and generated embeddings;
- *ranking score*: a percentage score measuring how many other test instances have a higher cosine similarity with a generated embedding than its reference does.

The definition modelling performance is measured by three too:

- *sense-BLEU*: sentence-BLEU implemented in NLTK with smoothing method 4 (Papineni et al., 2002; Chen and Cherry, 2014);
- *lemma-BLEU*: the maximum sense-BLEU between a generated gloss and all possible references of the same word and part of speech;
- *MoverScore*: a neural distance measure based on multilingual BERT (Zhao et al., 2019).

Finally, participants are ranked by rank scores instead of scalar numbers from the above metrics. A rank score is simply the rank of a particular submission among all submissions. For each sub-track, the average rank score of all three metrics is used to rank each team.

3 System Overview

3.1 Model Architecture

We select Chen and Zhao (2022)'s model as our system architecture because it has demonstrated great success on previous datasets for reverse dictionary and definition modelling. It is a "unified" model as it learns both tasks simultaneously, based on the intuition that a word and its corresponding definition share the same meaning, thus can be cast into the same neural semantic space.

We attach a diagram of this architecture as Figure 1. Technically, the model encodes glosses or word embeddings as the input, maps it into a shared representation, then generates embeddings or glosses accordingly. The shared representation serves as an autoencoding of both a word and its definition. Specifically, Linear layers (L) transform embeddings, and Transformer (Vaswani et al., 2017) blocks (T) encode or decode definitions.

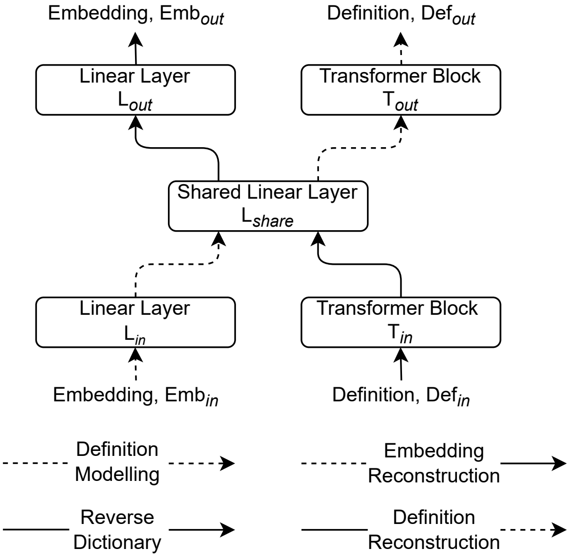


Figure 1: Chen and Zhao's illustration of the unified model.

3.2 Multi-task training

At the bottom of Figure 1, four trainable objectives are depicted: definition modelling, reverse dictionary, along with word embedding and definition reconstruction. The first two are CODWOE tasks, and the rest are auxiliary autoencoding tasks. Besides, another objective is to bring the vector representations of a word and its definition close in the shared layer. Our overall objective function combines the five objectives with equal weights.

3.3 Ensembling for reverse dictionary

Ensembling is a commonly employed technique to enhance machine learning performance. Specifically for reverse dictionary, we perform average ensembling: for each test instance, its final prediction is obtained by averaging all the corresponding predictions from different models. We ludicrously ensemble up to 21 models, of the same unified architecture, trained with various random seeds.

3.4 Handcrafting for definition modelling

Upon our initial inspection of definition modelling on the trial set, the generated definitions are mostly meaningless hallucinations, scoring a very low sense-BLEU of about 3. To understand how indicative BLEU is in this case, we handcraft a nonsensical n-gram submission. The rule is that for each test instance, we simply concatenate the most frequent bigram with the most frequent unigram, computed on all definitions in the training data. The phrases we prepare for each language are:

en	es	fr	it	ru
, or .	de la .	) (.	) (.	в . ,

4 Experiments and Results

4.1 Experimental setup

We tokenize glosses by whitespaces, add tokens into an open vocabulary, and embed them using one-hot. Word embeddings are used as provided. Loss functions are cross-entropy for tokens and MSE for embeddings. We also try cosine similarity for embeddings, but the model fails to converge. For definition modelling, we do not combine various embeddings as the input; this might put us at disadvantage in the team ranking.

While Transformer components are connected to form a unified model, most hyperparameters remain the same as in the provided baseline, which we specify in Appendix A. Following the original work, we tie Transformer embeddings and add a residual connection. We follow the same configurations for all language-embedding combinations. Training a unified model on an Nvidia GeForce RTX 2080 Ti takes roughly three hours.

4.2 Results

During the evaluation, we submit the provided baseline and our unified model. Also, we add ensembles of 17 and 21 models, as well the handcrafted n-grams. The submission scores, computed by the task organizers, are reported in Table 2 and 3. In the direction of reverse dictionary, the unified model steadily beats the baseline; ensembling adds a cherry on top for some languages but not all.

In definition modelling, our n-grams surpass genuine models on *en* BLEU scores, and even rank first in *fr* sense-BLEU among all participants' entries. This implies that either BLEU scores are not informative, or the model outputs are as embarrassing as the n-grams. On contrary, MoverScore is effective in downing the n-grams, probably by penalizing disfluency or semantic mismatch. Sadly, our manual review suggests that most model-generated

	en			es			fr			it			ru		
	MSE	cosine	rank	MSE	cosine	rank	MSE	cosine	rank	MSE	cosine	rank	MSE	cosine	rank
baseline	0.884	0.189	0.439	0.905	0.241	0.462	1.06	0.275	0.360	1.10	0.245	0.451	0.561	0.295	0.432
unified	0.871	**0.241**	**0.326**	0.868	0.339	**0.271**	**1.03**	**0.312**	**0.302**	1.05	0.371	**0.197**	0.553	0.327	0.340
ensemble 17	**0.864**	0.225	0.374	**0.860**	**0.347**	**0.271**	**1.03**	0.305	0.334	**1.03**	0.373	0.206	**0.538**	0.381	0.251
ensemble 21	0.865	0.225	0.374	**0.860**	**0.347**	**0.271**	**1.03**	0.306	0.330	**1.03**	**0.374**	0.205	**0.538**	**0.383**	**0.247**

(a) *sgns* as target embeddings

	en			es			fr			it			ru		
	MSE	cosine	rank	MSE	cosine	rank	MSE	cosine	rank	MSE	cosine	rank	MSE	cosine	rank
baseline	0.161	0.795	0.500	0.551	0.820	0.499	0.404	0.764	0.495	0.400	0.720	0.499	0.144	0.829	0.496
unified	0.143	0.795	0.500	0.480	0.834	0.431	0.347	0.782	0.448	0.337	0.745	**0.428**	0.119	0.849	0.395
ensemble 17	**0.142**	0.795	0.500	**0.467**	**0.839**	**0.424**	0.336	0.788	0.429	**0.334**	**0.747**	0.429	**0.116**	0.851	0.390
ensemble 21	**0.142**	0.795	0.500	**0.467**	**0.839**	0.425	**0.335**	**0.789**	**0.428**	**0.334**	**0.747**	0.429	**0.116**	**0.852**	**0.389**

(b) *char* as target embeddings

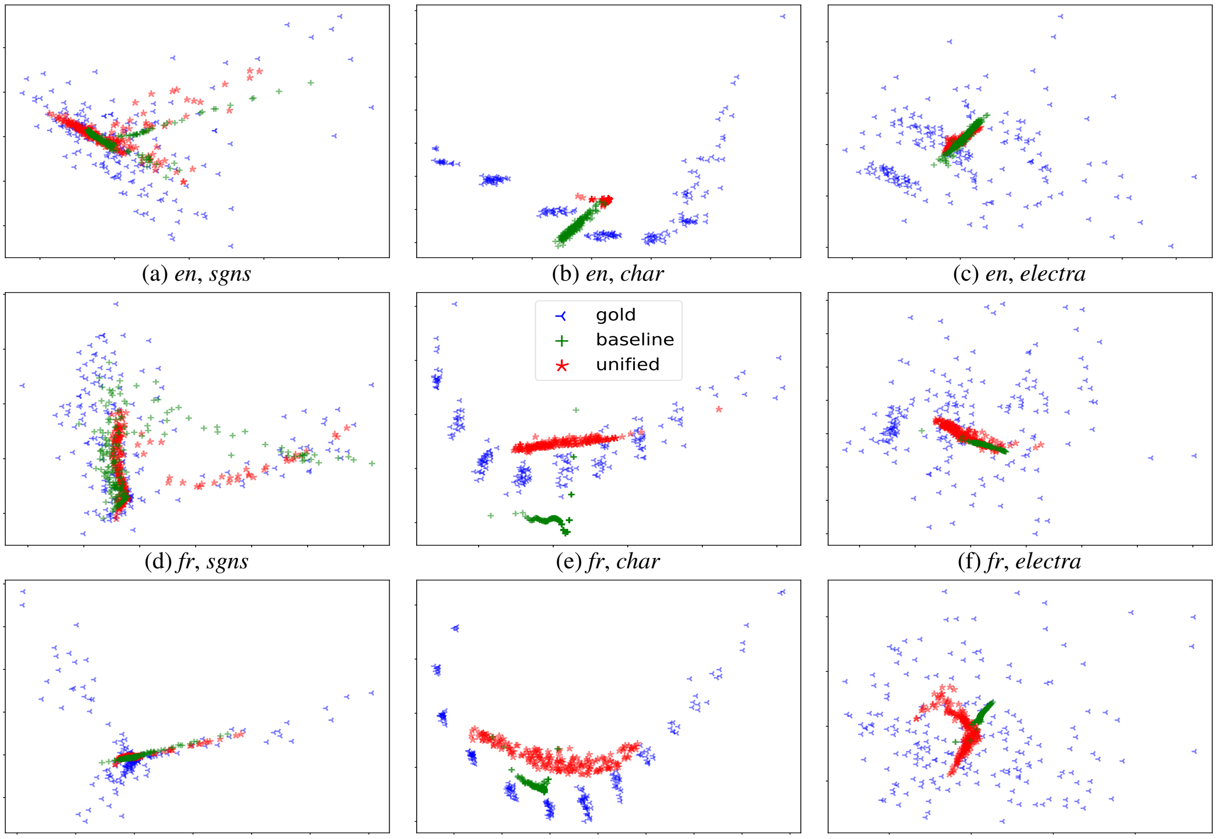
	en			fr			ru		
	MSE	cosine	rank	MSE	cosine	rank	MSE	cosine	rank
baseline	1.34	0.842	0.497	1.18	0.853	0.497	0.898	0.718	0.498
unified	1.32	0.844	0.495	1.08	0.861	0.476	0.846	0.731	0.421
ensemble 17	**1.31**	**0.847**	**0.490**	**1.07**	**0.862**	**0.479**	**0.829**	**0.735**	**0.417**
ensemble 21	**1.31**	**0.847**	0.491	**1.07**	0.861	0.480	**0.829**	0.734	0.419

(c) *electra* as target embeddings

Table 2: Reverse dictionary test performance, measured by MSE (↓), cosine similarity (↑), and ranking score (↓).

	source	en			es			fr			it			ru		
	embed.	MvSc	s-B	l-B	MvSc	s-B	l-B	MvSc	s-B	l-B	MvSc	s-B	l-B	MvSc	s-B	l-B
n-grams	n/a	-0.004	**3.06**	**3.81**	-0.032	2.73	3.67	-0.176	**2.95**	3.56	-0.164	1.89	2.74	-0.006	2.65	3.31
baseline	sgns	0.100	2.91	3.67	0.088	**3.47**	**5.28**	-0.019	2.34	3.38	0.046	4.62	6.97	**0.109**	**4.91**	7.14
unified		0.098	3.01	3.80	**0.101**	3.42	5.14	-0.064	1.59	2.38	**0.107**	**6.01**	**9.17**	0.095	4.59	6.82
baseline	char	0.101	2.47	3.02	0.064	2.06	2.88	-0.186	0.11	0.11	0.019	2.09	2.99	0.092	4.01	5.87
unified		**0.104**	2.83	3.40	0.065	2.14	2.96	**0.026**	2.42	**3.82**	0.044	2.93	4.29	0.085	4.80	**7.24**
baseline	electra	0.070	2.53	3.26	n/a			-0.075	1.38	1.93	n/a			0.090	3.78	5.45
unified		0.094	2.75	3.43				-0.045	1.60	2.29				0.088	4.08	5.86

Table 3: Definition modelling test results, in MoverScore (↑), sense-BLEU (↑), and lemma-BLEU (↑).



(a) *en*, *sgns*
(b) *en*, *char*
(c) *en*, *electra*
(d) *fr*, *sgns*
(e) *fr*, *char*
(f) *fr*, *electra*
(g) *ru*, *sgns*
(h) *ru*, *char*
(i) *ru*, *electra*

Figure 2: Visualization of gold and output embedding distributions across languages and embedding architectures.

glosses are inaccurate. The dissatisfying results might be due to the modest training data size.

5 Performances across *embeddings*

Reverse dictionary MSE and cosine are incomparable across different embedding types, whereas ranking scores can tell which embedding architecture is preferred for indexing and retrieving a word. A random baseline ranking score is 0.5, and most *char* and *electra* figures, unfortunately, fall between 0.4 and 0.5. On the other hand, *sgns* is more useful as its baseline scores start at around 0.45, and our models can improve these up to 0.25.

We employ principal component analysis (PCA) to reduce the gold and output embeddings to 2 dimensions. Then in Figure 2 we visualize *en*, *fr*, and *ru*, which come with all embeddings. The unified model usually outputs to a larger space than the baseline, hinting at a positive correlation between output spread and performance. Gold *electra* has the most isotropic space, but neither model could imitate the distribution. *Char* has a crescent shape with several clusters inside, which is unlikely to be cosine-friendly. These problems are alleviated on *sgns*, which witnesses the best ranking scores.

Definition modelling *Sgns* is again the winner, as models trained with it reach the top in many metrics. *Char* is also favourable. This is counter-intuitive as *electra* should be fitter, for it retains more sense-specific knowledge. A possible reason is that *electra* needs to go through more training data than *sgns* and *char* to reach perfection.

6 Performances across *languages*

As seen in the result and rank tables, our system's behaviour is relatively consistent on various languages, except that English is more challenging. Assuming that the datasets are of similar quality, it is questionable to conclude that our model suits other languages more than English. Moreover, Figure 2 confirms that the English embeddings are not more peculiar than those of other languages.

We guess that other teams have focused on English (e.g. only submitted English), as it is a centred language in the research community. Instead, our hyperparameter search is based on the average loss from all languages, neglecting that the losses are not directly comparable.

7 Performances across *linguistic features*

We look into the unified model's trial set predictions, to interpret how scores vary across diverse linguistic annotations: polysemy, part of speech (POS), word length in characters, definition length in words, and word frequency. For categorical features, we group data by annotations; for numerical features, we divide the data into three subsets, by percentile ranges: 0-33, 33-67, and 67-100. Statistics of the subsets are in Table 4. We list cosine similarity for reverse dictionary, and lemma-BLEU for definition modelling. A generic discovery is that, the best scores of the two tracks emerge in differing subsets, regardless of what the feature is.

Linguistic feature	**Category / Range**	**No. of instances**
Polysemy	Yes	65
	No	135
Part-of-speech	Adj	56
	Adv	11
	Verb	37
	Noun	96
Word frequency (frequency rank in the whole corpus)	67 – 11145	67
	11146 – 44416	66
	44417 – 905726	67
Word length	3 – 5	85
	6 – 7	60
	8 – 17	55
Definition length	1 – 6	71
	7 – 10	65
	11 – 39	64

Table 4: Statistics of the different subsets grouped by features.

Polysemy Table 5 exhibits the results for the words with either one or multiple definitions. It is slightly easier to achieve better cosine similarity for unambiguous words. Polysemous words have better BLEU, and *electra* has worse BLEU than *sgns*. This is illogical, as defining a polysemous word is harder, especially without context. We hypothesize that BLEU is not reflective, and *electra* embeddings might be of sub-optimal quality.

Polysemy	sgns		char		electra	
	cosine	l-B	cosine	l-B	cosine	l-B
Yes	0.232	**4.34**	0.804	**3.20**	0.836	**3.61**
No	**0.360**	2.82	**0.813**	2.53	**0.845**	3.09

Table 5: Performances across polysemy annotations for *en*.

Part of speech Next, numbers for the four POS tags that exist in *en* trial, are laid out in Table 6. Strong cosine similarity is associated with verbs, although cosine numbers are close, except for adverbs. Adverbs, which have a small sample size, dominate high lemma-BLEU, perhaps because they are the least ambiguous.

POS	sgns		char		electra	
	cosine	l-B	cosine	l-B	cosine	l-B
Adj	0.319	3.36	0.801	2.76	0.811	2.81
Adv	0.134	**6.56**	0.798	**5.45**	0.815	**5.93**
Verb	**0.383**	3.20	**0.839**	2.50	0.853	3.83
Noun	0.314	2.97	0.806	2.53	**0.860**	2.99

Table 6: Performance across POS tags for *en*.

Word length We then make three partitions according to different word length ranges. Results in Table 7 suggest that shorter words have higher cosine, while longer words have higher lemma-BLEU. Numbers are closer for *sgns* and *electra*; we further investigate on *char* in Section 8.1.

Word length	sgns		char		electra	
	cosine	l-B	cosine	l-B	cosine	l-B
short	**0.332**	3.19	**0.845**	2.58	0.817	3.10
medium	0.314	3.19	0.842	2.74	**0.867**	**3.41**
long	0.327	**3.66**	0.694	**3.00**	0.854	3.33

Table 7: Performances across word lengths for *en*.

Definition length Likewise in Table 8, we separate the trial data by the gold definition length. Much higher BLEU is seen when the model defines words linked with a shorter gold gloss, as generating a shorter sequence is easier. As we anticipate, when the model produces word embeddings for longer glosses, results are better too, potentially because more information can be encoded.

Definition length	sgns		char		electra	
	cosine	l-B	cosine	l-B	cosine	l-B
short	0.280	**4.51**	0.796	**3.60**	0.824	**4.89**
medium	0.318	3.48	0.814	2.73	0.848	2.76
long	**0.361**	1.83	**0.822**	1.80	**0.856**	1.93

Table 8: Performances across definition lengths for *en*.

Word frequency Finally, Table 9 summarizes the results of the low, medium, and high frequency word groups. From the results, we cannot establish an explicit trend across different task directions, embeddings, or word frequencies. This implies that the embedding quality and model performance might be word frequency-agnostic.

Frequency	sgns		char		electra	
	cosine	l-B	cosine	l-B	cosine	l-B
low	0.250	3.53	0.805	**2.82**	0.850	3.30
medium	0.348	**3.54**	0.786	2.76	**0.864**	**3.38**
high	**0.357**	2.89	**0.839**	2.66	0.814	3.10

Table 9: Performances across word frequencies for *en*.

8 Qualitative Analysis and Discussions

8.1 Observing the crescent with a telescope

After PCA retains the most distinguishing components, Figure 2 shows interesting patterns, especially for *char*. We randomly label 25 English words and present them in Figure 3 and Figure 4, respectively for *char* and *electra*. The sub-clusters in *char*'s crescent are perfectly in tune with word lengths; for *electra*, more frequent words are closer to the origin. We do not notice a clear trend for *sgns*, for which a plot is attached as Figure 5.

We attribute the distinct patterns to the training paradigms: character-level word autoencoding for *char*, and contextualized modelling for *electra*. This accounts for the largest cosine gap on *char* between long and short words, seen earlier in Table 7. Intuitively, it is more difficult to train *char* autoencodings for longer words, so, in turn, embeddings for longer words possess inferior quality.

Within *char* embeddings, words are grouped by lengths, so we may utilize this for word retrieval in future work. Nonetheless, we are unsure of how length or frequency information aids sense-based tasks, like definition generation in our context.

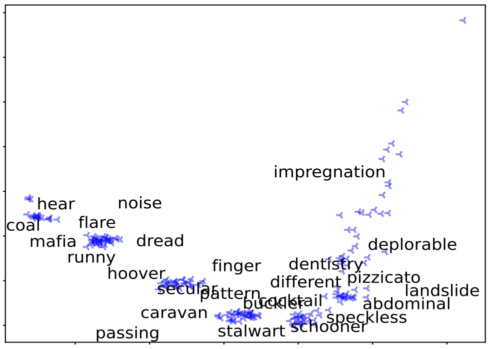


Figure 3: Gold English *char* embeddings with word labels.

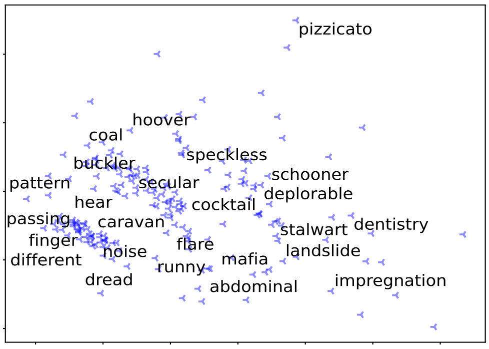


Figure 4: Gold English *electra* embeddings with word labels.

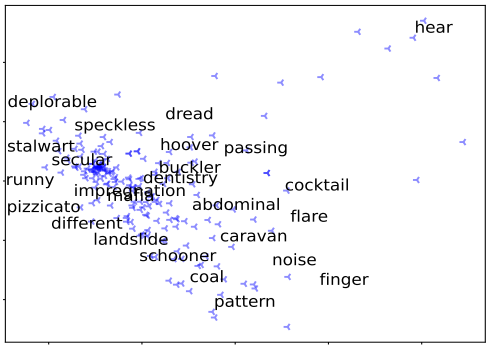


Figure 5: Gold English *sgns* embeddings with word labels.

8.2 Sense-BLEU with no sense

We design a sanity check on the representativeness of BLEU. On the English trial set, we remove punctuation marks and NLTK-defined stop words from both references, and our unified model's definitions generated from *sgns*. Sense-BLEU drops from 3.31 to 0.39, and surprisingly, it worsens to 0 with smoothing disabled. Evidently, sense-BLEU and thereby lemma-BLEU are hugely inflated by functional tokens as well as smoothing.

8.3 Evaluating task evaluation and ranking

We point out the limitations associated with the evaluation and ranking process, which can benefit from a rethink. First, as shown above, the two BLEU metrics may not be practical. Second, some metrics are correlated, i.e., cosine with the ranking score, and sense-BLEU with lemma-BLEU. These problems are amplified by the team ranking protocol, which averages a team's ranks in individual metrics to produce a final standing. It might not be meaningful to compare the individual metric ranks, not to mention averaging them since metrics are not equally weighted.

Nonetheless, we are not in a knowledgeable position to propose a better approach, other than clumsily displaying ranks in individual metrics.

Acknowledgements

Pinzhen Chen is supported by funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825303 (Bergamot). Zheng Zhao is supported by the UKRI Centre for Doctoral Training in Natural Language Processing (UKRI grant EP/S022481/1). This paper reflects the view of the authors and not necessarily that of the funders.

References

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

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level BLEU. In *Proc. of WMT*.

Pinzhen Chen and Zheng Zhao. 2022. A unified model for reverse dictionary and definition modelling. *arXiv*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *ICLR*.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to understand phrases by embedding the dictionary. *TACL*, 4:17–30.

Timothee Mickus, Denis Paperno, Mathieu Constant, and Kees van Deemter. 2022. SemEval-2022 Task 1: CODWOE – comparing dictionaries and word embeddings. In *Proc. of SemEval*.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*.

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 *NeurIPS*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proc. of EMNLP-IJCNLP*.

A Hyperparameters

Hyperparameter	**Value**
learning rate	1e-4
optimizer	Adam
beta1, beta2	0.9, 0.999
weight decay	1e-6
batch size	256
decoding beam size	6
early stopping	5 non-improving validations
embedding loss	mean squared error
token loss	cross-entropy
Transformer depth	4
Transformer head	4
Transformer dropout	0.3
linear dropout	0.2
shared layer dim.	256
word embed.	*sgns*, *char*, *electra*
word embed. dim.	256
definition embed.	one-hot
definition embed. dim.	256
vocabulary size	open, all training tokens

Table 10: Model hyperparameters.