Semeval-2022 Task 1 : CODWOE -- Comparing Dictionaries and Word Embeddings

Mickus, Timothee

The Association for Computational Linguistics
2022-07

Mickus , T , Van Deemter , K , Constant , M & Paperno , D 2022 , Semeval-2022 Task 1 : CODWOE -- Comparing Dictionaries and Word Embeddings . in G Emerson , N Schluter , G Stanovsky , R Kumar , A Palmer , N Schneider , S Singh & S Ratan (eds) , Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022) . The Association for Computational Linguistics , Stroudsburg , pp. 1-14 , International Workshop on Semantic Evaluation , Seattle , Washington , United States , 14/07/2022 . https://doi.org/10.18653/v1/2022.semeval-1.1

http://hdl.handle.net/10138/348397
https://doi.org/10.18653/v1/2022.semeval-1.1

Semeval-2022 Task 1: CODWOE – Comparing Dictionaries and Word Embeddings

Timothee Mickus*
Helsinki University
timothee.mickus@helsinki.fi

Kees van Deemter
Utrecht University
c.j.vandeemter@uu.nl

Mathieu Constant
Université de Lorraine
CNRS, ATILF
mconstant@atilf.fr

Denis Paperno
Utrecht University
d.paperno@uu.nl

Abstract

Word embeddings have advanced the state of the art in NLP across numerous tasks. Understanding the contents of dense neural representations is of utmost interest to the computational semantics community. We propose to focus on relating these opaque word vectors with human-readable definitions, as found in dictionaries. This problem naturally divides into two subtasks: converting definitions into embeddings, and converting embeddings into definitions. This task was conducted in a multilingual setting, using comparable sets of embeddings trained homogeneously.

1 Introduction

Word embeddings are a success story in NLP. They have been equated to distributional semantics models (Lenci, 2018; Boleda, 2020), a theory of semantics which relates the meaning of words to their distribution in context (Harris, 1954). Recently introduced contextualized word embeddings (e.g. Devlin et al., 2019) have set a new state of the art on a wide variety of tasks. For this reason, they have attracted much research interest. Do they depict consistent semantic spaces and are they theoretically valid (Mickus et al., 2020b; Yenicelik et al., 2020)? What limitations are to be expected in these models (Bender and Koller, 2020)? Can they scale up in performance (Brown et al., 2020)?

Word embeddings are dense vector representations of meaning which are not easily intelligible to a human observer. Many techniques have been employed to make embedding spaces more interpretable. A promising approach consists in *converting these opaque vectors into human readable definitions, as one could find in a dictionary*: accurately translating a dense, opaque vector representation into an equivalent human-readable piece of text would allow us to peer into the black box of modern neural network architectures. This avenue of research, known as definition modeling, was pioneered by Noraset et al. (2017). One may however question whether the task is at all feasible: there is no guarantee that the information content of a dictionary definition is similar to that which is described by real-valued vectors inferred from word distributions.

Figure 1: Logo for CODWOE shared task

The SemEval Shared Task on Comparing Dictionaries and Word Embeddings (CODWOE) sets out to study whether embeddings and dictionaries encode similar information. We present the task and relevant state of the art in Section 2. We describe the data collected and presented to participants in Section 3. In Section 4, we discuss the metrics used to rank participant submissions. Our baseline model is presented in Section 5. We list results from participants' submissions in Section 6 and provide a more in-depth discussion in Section 7.

2 What we are fishing for

What is in a word embedding? Are word embeddings semantic descriptions, in the same sense that

*Work conducted while at ATILF

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1 - 14

dictionary definitions are? If so, embeddings and definitions must be translatable into one another. The CODWOE shared task was set up to test this. The shared task participants investigated whether a word vector—e.g. $\vec{\texttt{cod}}$—contains the same information as the corresponding dictionary definition—viz. "*any of various bottom-dwelling fishes (family Gadidae, the cod family) that usually occur in cold marine waters and often have barbels and three dorsal fins*."[1]

We decompose this research problem into two tracks: the first corresponds to the vector-to-sequence task of Definition Modeling, the second to the sequence-to-vector Reverse Dictionary task. The task of definition modeling consists in using the vector representation of $\vec{\texttt{cod}}$ to produce the associated gloss, "*any of various bottom-dwelling fishes (family Gadidae, the cod family) that usually occur in cold marine waters and often have barbels and three dorsal fins*". The reverse dictionary task is the mathematical inverse: reconstruct an embedding $\vec{\texttt{cod}}$ from the corresponding gloss.

These two tracks display a number of interesting characteristics. These tasks are obviously useful for explainable AI, since they involve converting human-readable data into machine-readable data and back. They also have a theoretical significance: both glosses and word embeddings are representations of meaning, and therefore involve the conversion of distinct non-formal semantic representations. From a practical point of view, the ability to infer word-embeddings from dictionary resources, or dictionaries from large un-annotated corpora, would prove a boon for many under-resourced languages.

2.1 Track 1: Definition Modeling

The first track consists in an application of Definition Modeling. As training material, participants have access to a set of data points, each of which consists of a source word embedding and a corresponding target word definition (see Figure 2). Participants are tasked with generating new definitions for an unseen test set of embeddings.

Definition Modeling is a recent addition in NLG tasks (Noraset et al., 2017) which seeks to do just that. It has since then gained traction (Gadetsky et al., 2018; Mickus et al., 2019; Li et al., 2020; Zhang et al., 2020a, a.o.). Other languages than English have also been studied, including Chinese (Yang et al., 2019), French (Mickus et al., 2020a), Wolastoqey (Bear and Cook, 2021), and more (Kabiri and Cook, 2020). At its very inception, Definition Modeling was suggested as a means of evaluating the content of distributional semantic models (Noraset et al., 2017). In practice however, different researchers rarely use comparable sets of embeddings (Mickus et al., 2020a), effectively making proper comparisons across systems impossible as they use distinct inputs. To fill this gap, we created a dataset of comparable embeddings from different languages and neural architectures, trained as homogeneously as possible on comparable data; see 3.2 below.

2.2 Track 2: Reverse Dictionary

Reverse dictionaries (a.k.a. retrograde dictionaries) are lexical resources that flip the usual structure of dictionaries, allowing users to query words based on the definitions they would expect them to have. One of the major challenges of such resources consists in providing definition glosses that match with users' expectations. As a consequence, a trend of research in NLP has focused on producing dynamic reverse dictionaries, that would interpret input definitions and map them back to the corresponding word. We refer the reader to the comprehensive review of Siddique and Sufyan Beg (2019), and provide here mainly highlights.

An early strand of research focused on augmenting definitions using synonyms or other semantically related words, such as hypernyms or hyponyms. This approach has been applied to multiple languages, from Turkish to English and to Japanese (Shaw et al., 2013; Bila et al., 2004; El Khalout and Oflazer, 2004). Building on this query-augmentation approach, we find works focused on integrating richer lexical resources, such as WordNet, the Oxford dictionary, The Integral Dictionary, or LDA vector spaces (Dutoit and Nugues, 2002; Thorat and Choudhari, 2016; Méndez et al., 2013; Calvo et al., 2016).

A related trend of research is that of Zanzotto et al. (2010) and Hill et al. (2016), who use dictionaries as benchmarks for compositional semantics. Zanzotto et al. (2010) used a shallow neural network to implement a compositional distributional semantics model and use dictionaries as their training data. Hill et al. (2016) instead employ a LSTM to parse the full definition gloss and use the hidden state at the last time-step to predict the word be-

[1] From Merriam-Webster.

ing defined. In both cases, replacing the definition gloss with a user's query would lead to a reverse dictionary system. Since then, a number of works have attempted to implement reverse dictionaries using neural language models. The WantWords system (Zhang et al., 2020b; Qi et al., 2020) is based on a BiLSTM architecture, and incorporates auxiliary tasks such as part-of-speech prediction to boost the performance. Yan et al. (2020) seeks to replace the learned neural language models in Hill et al. (2016) or WantWords with a pre-trained model such as BERT (Devlin et al., 2019) and its multilingual variants, which allows them to use their system in a cross-lingual setting—querying in a language to obtain an answer in another. Most recently, Malekzadeh et al. (2021) used a neural language model based approach to implement a Persian reverse dictionary.

With respect to the CODWOE shared task, our interest lies in reconstructing the word embedding of the word being defined, rather than finding the corresponding word—an approach more closely related to that of Zanzotto et al. (2010) and Hill et al. (2016). Under this slight reformulation, the sequence-to-vector Reverse Dictionary task is strictly the inverse of the vector-to-sequence task of Definition Modeling. Hence we define the Reverse Dictionary task as *computing the components of a target word vector using as input a human-readable definition*. To solve this task, participants have access to a set of data points, each of which consists of a source word definition and a corresponding target word embedding, as training materials.

3 What's in the nets: Data used

The definition modeling and reverse dictionary tasks both require a parallel dataset, where dictionary definitions are aligned with corresponding word embeddings. The task is held in a multilingual setting. We provide data in English, French, Russian, Italian and Spanish. We selected these languages to facilitate the collection of comparable data: all these languages possess comparable large scale resources, including online dictionaries as well as corpora that can be used to train comparable embeddings. Our datasets are made available online at `https://codwoe.atilf.fr/`.

The aim of both tracks of CODWOE is to compare the semantic contents of definitions and embeddings. As a consequence, we ask participants to refrain from using external data such as pretrained models and lexical resources: including such external data would introduce another source of semantic information, and obfuscate the results from this shared task.

	with examples	without
en	0	806297
es	0	132583
fr	431793	573313
it	16127	86959
ru	122282	485208

Table 1: DBnary: number of items per language

	N. Sents.	**N. Tokens**	**N. Bytes**
it	78761031	955474050	5001829910
es	78973969	975762257	5001999992
fr	82082118	1004767254	5001999368
en	97622760	1035154295	5001999755
ru	79526583	1035661601	10036395727

Table 2: Embeddings: corpus statistics

3.1 Dictionary data

As a source of dictionary definitions, we primarily use the DBnary dataset (Sérasset, 2012),[2] an RDF-formatted version of some of the existing Wiktionary projects.[3] DBnary includes data for all of our selected languages. One sub-dataset per language is constructed. Definitions are selected according to corpus frequency and part-of-speech of the word being defined. We solely select nouns, adjectives, verbs and adverbs.

Table 1 presents the number of usable items in DBnary. Not all languages contain examples of usage. A brief regular expression lookup suggests that around 20K examples of usage can be found in the Spanish version of Wiktionary, while English yields at least 200K. We therefore discard the English version of DBnary and replace it by a manual parse, from which we also retrieve examples of usage.

3.2 Embeddings data

We have collected similar amounts of data for each language (Table 2) to use as training corpora. The sources we use to constitute these corpora are selected to be generally comparable: each cor-

[2] `http://kaiko.getalp.org/about-dbnary/`
[3] See `https://www.wiktionary.org/`

pus contains 2.5G data parsed and cleaned from Wikipedia,[4] 2.2G from the OpenSubtitles OPUS corpus (Lison and Tiedemann, 2016),[5] as well as 0.3G in books from various genres, drawn from LiberLiber[6] for Italian, Wikisource for Spanish and Russian, and Gutenberg[7] for English and French.

We focus on three embedding architectures: word2vec models (Mikolov et al., 2013) trained with gensim (Řehůřek and Sojka, 2010), the ELECTRA model of Clark et al. (2020), and character-based embeddings. The word2vec and ELECTRA models were selected so as to provide some comparison between static and contextual embeddings; both are trained with default hyperparameters aside from output vector size, which we set to 256. As for the ELECTRA models, given that we need contexts to derive token representations, we train the models only in English, French and Russian. The Spanish and Italian Wiktionary projects contain too few examples of usage. For French and Russian, we derive contextualized embeddings of a word to be defined from usage examples in DBnary datasets. Since the English DBnary dataset does not contain examples of usage, we extracted them from the original Wiktionary dumps.

The character-based embeddings are included to provide baseline expectations for non-semantic representations—as we can expect spelling to be more or less arbitrary with respect to word meaning (Saussure, 1916).[8] In practice, these embeddings are computed through a simple LSTM-based auto-encoder: the word is passed into an LSTM encoder as a sequence of characters, we sum all output hidden states, and use these summed hidden states to initialize an LSTM decoder, whose objective is to reconstruct the input word. As a character-based representation, we can therefore use the summed output hidden states, as they are tailored to contain all the information necessary to reconstruct the spelling of the corresponding word.[9] The datasets used to trained the models correspond to the set of all word types attested in our base corpora described in Table 2. All models achieve a 99% reconstruction accuracy.

word	POS	gloss
sminuire	V	far figurare qualcosa o qualcuno come meno importante o rilevante

(a) Example definition in Italian

```
{
    "id": "it.42",
    "word": "sminuire"
    "gloss": "far figurare...",
    "pos": "v",
    "electra": [0.4, 0.2, ...],
    "sgns": [0.2, 0.4, ...],
    "char": [0.3, 1.4, ...],
}
```

(b) Corresponding JSON snippet

Figure 2: Toy example data point in the Italian dataset

3.3 Datasets

We construct one dataset per language. Each language-specific dataset is split in five: a trial split (200 datapoints per language), a training split (43 608 datapoints), a validation split (6375 datapoints), a definition modeling testing split (6221 datapoints) and a reverse-dictionary testing split (6208 datapoints). Splits are constructed such that there are no overlap in the embeddings. Dataset splits are formatted as JSON files.

Each file consists of a list of JSON dictionary notations. JSON items contain a unique identifier for the data point, the word being defined, definition, part of speech, and all word vectors. A depiction of the sort of items included in our datasets is shown in Figure 2. Sub-figure 2a summarizes the data presented as a JSON item in Sub-figure 2b.

Participants had access to the trial, train and validation splits of all languages. Test splits were made available at the beginning of the evaluation period.

4 The scales we use

We now turn to the metrics of our shared task.

[4] See here: https://dumps.wikimedia.org/

[5] See https://opus.nlpl.eu/

[6] Cf. https://www.liberliber.it/online/

[7] See here: https://www.gutenberg.org/

[8] Nonetheless, see Gutiérrez et al. (2016), Kutuzov (2017), Dautriche et al. (2017) or Pimentel et al. (2019), all of which question this assumption.

[9] Given that we implement this module ourselves, we use a Bayesian Optimization algorithm (Snoek et al., 2012) to select hyperparameters for our five character auto-encoder. We use this process to decide learning rate, weight decay, dropout, β_1 and β_2 parameters of the AdamW optimizer, batch size, number of epochs over the full dataset, as well as whether to share a single weight matrix for encoder and decoder character embeddings.

4.1 Reverse Dictionary Metrics

The Reverse Dictionary task, as we have re-framed it here, consists in reconstructing embeddings. To that end, we consider three measures of vector similarity. First is MSE (mean squared error), which measures the difference between the components of the reconstructed and target embeddings. Mean-squared error is however not very easy to interpret on its own. Second is cosine: the reconstructed and target embeddings should have a cosine of 1. It is hard to place specific expectations for what a random output would produce, as this essentially differs from architecture to architecture: for instance, Transformer outputs are known to be anisotropic, so we shouldn't expect two random ELECTRA embeddings to be orthogonal (Ethayarajh, 2019; Timkey and van Schijndel, 2021, a.o.).

As neither MSE nor cosine provides us with a clear diagnosis tool comparable across all targets, we also include a ranking based measure: we compare the cosine of the reconstructed embedding $\vec{p}_i$ and the target embedding $\vec{t}_i$ to the cosine of the reconstruction $\vec{p}_i$ and all other targets $\vec{t}_j$ in the test set, and evaluate the proportion of such targets that would yield a closer association—viz., the number of cosine values greater than $\cos(\vec{p}_i, \vec{t}_i)$. More formally, we can describe this ranking metric as:

$$\text{Ranking}(\vec{p}_i) = \frac{\sum_{\vec{t}_j \in \text{Test set}} \mathbb{1}_{\cos(\vec{p}_i,\vec{t}_j) > \cos(\vec{p}_i,\vec{t}_i)}}{\#\text{Test set}} \quad (1)$$

4.2 Definition Modeling Metrics

A common trope in NLG is to stress the dearth of adequate automatic metrics. Most of the metrics currently existing focus on token overlap, rather than semantic equivalence. The very popular BLEU and ROUGE metrics (Papineni et al., 2002; Lin, 2004) measure the overlap rate in n-grams of various lengths (usually 1-grams to 4-grams).

To alleviate this, researchers have suggested using external resources, such as lists of synonyms and stemmers (Banerjee and Lavie, 2005) or pre-trained language models (Zhao et al., 2019). The reliance of these augmented metrics on external resources is problematic. Different languages will use different resources with varying degrees of quality—and this will necessarily impact scores, introducing a confounding factor for any analysis down the line. In the extreme case, if these resources are not available for a particular language, then the metric will have to be discarded. Even assuming the availability of the required external resources, none of these improved metrics is entirely satisfactory. In the case of synonymy-aware metrics such as METEOR (Banerjee and Lavie, 2005), we can stress that syntactically different sentences can express the same meaning, but would not be captured by such metrics. Embeddings-based metrics such as MoverScore (Zhao et al., 2019) are very recent, and therefore less well understood; moreover concerns can be raised about whether using a method derived from neural networks trained on text will prove of any help in studying the meaning of texts generated by other neural networks.

One alternative frequently used by the NLG community is perplexity, which weighs the probability that the model would generate the target. This last alternative is however not suited to a shared task setup, as it requires us to have access to the actual neural networks trained by participants so we can investigate the probability distributions they model—unlike the other metrics we mentioned thus far, which only require model outputs.

In short, none of the currently available NLG metrics are fully satisfactory. Some are not applicable given the shared task format, some depend on external resources of varying quality, and some merely measure formal similarity, rather than semantic equivalence. Our approach is therefore twofold: on the one hand, we select multiple metrics with the expectation that each might shed light on one specific factor; on the other hand, we encourage participants to go beyond automatic scoring for the evaluation of their model.

As for which metrics we select, we narrow our choice to three. First is a basic BLEU score (Papineni et al., 2002) between a production p_i and the associated target t_i; our reasoning here is that as it is one of the most basic metrics, it is a consistent default choice. Second is the maximum BLEU score between a production p_i and any of the targets $t_i, t_j \ldots t_n$ for which the definiendum is the same as that of p_i. This second metric is designed to not penalize models that rely solely on SGNS or char embeddings: as the input would always be the same, deterministic models would always produce the same definition $p_i = p_j = \ldots = p_n$.[10] To distinguish between our two BLEU variants, we refer to the former as S-BLEU (or Sense-BLEU),

[10] One way of bypassing this problem would be to include a source of noise, as is done in GAN architectures (Goodfellow et al., 2014). This would still leave open the question of how to optimally align the outputs to the possible targets.

and the latter as L-BLEU (or Lemma-BLEU).

Given that some definitions in our dataset can be very short, we also apply a smoothing to both BLEU-based metrics. In practice, BLEU computes an overlap of n-grams of size m and under; by default, $m = 4$. This overlap is a geometric mean across all n-gram sizes $1 \dots m$. If a definition d contains less than m tokens, then any associated production for which d is used as a target will contain 0 overlapping n-grams of size m. The use of a geometric mean then entails that the BLEU score for any production associated to d will be 0. To circumvent this limitation of BLEU, it is common to use some form of smoothing. Here, for any n-gram size $\hat{m}$ that would yield an overlap of 0 (i.e., $\hat{m}$ such that $\#d < \hat{m} \leq m$), we replace the overlap count with a pseudocount of $1/\log \#d$.

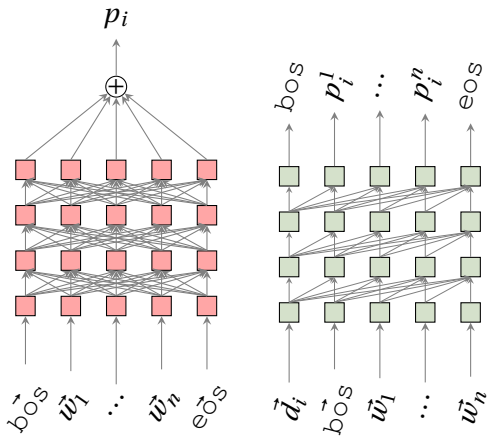
Lastly, we include MoverScore (Zhao et al., 2019), using a multilingual DistilBERT model as the external resource. The fact that this model is multilingual means that we can use it for all five languages of interest. Embedding-based methods have the potential to overcome some of the limitations of purely token-based metrics, which is why we deem them worth including in our setup.

The second part of our approach for evaluating submissions consists in encouraging participants to not rely solely on the automatic scoring system of their outputs. Concretely, we provide participants with a richly annotated trial dataset, which contains frequency and hand-annotated semantic information, and strongly suggest participants to use it for a manual evaluation of their system. We include the presence of a manual evaluation as a criterion to evaluate the quality of a system description paper, and plan to formally recognize the most enlightening evaluations conducted by participants.

Neither our selection of metrics nor our insistence on manual evaluation solves the evaluation issues of NLG systems. We duly note the importance of this question, and plan to conduct a follow-up evaluation campaign on the CoDWoE submissions.

5 Testing the waters: baseline architectures

We implement simple neural network architecture baselines to lower the barrier to entry to this shared task. They are based on the Transformer architecture of Vaswani et al. (2017) and designed to be as simple as possible. Our code is publicly available at `https://github.com/TimotheeMickus/codwoe`.



(a) Reverse dictionary (b) Definition modeling

Figure 3: Baseline architectures for the CoDWoE shared task

We illustrate our Reverse Dictionary baseline architecture in Figure 3a. It consists in feeding the input gloss $\langle \vec{\texttt{bos}}, \vec{w}_1, \dots, \vec{w}_n, \vec{\texttt{eos}} \rangle$ into a simple Transformer encoder, and then summing all the hidden representations to produce the prediction p_i. In practice, the summed hidden states are passed into a small non-linear feed-forward module to derive the prediction:

$$p_i = W_p \left(\text{ReLU} \left(\sum_t \vec{h}_t \right) \right) \quad (2)$$

Our Definition Modeling baseline is presented in Figure 3b. It consists in a simple Transformer encoder, where earlier time-step representations are prevented from attending to later time-step representations. To provide information about the definiendum to the model, we use the definiendum embedding $\vec{d}_i$ as the input for the first time-step instead of a start-of-sequence token. We train the models with teacher-forcing: i.e., during training we ignore the definientia $p_i^1, \dots, p_i^n$ that the model produces; instead we feed it the target $w_1, \dots, w_m$ attested in the training set at each time-step. During inference, we feed the model with its own prediction. This creates a train-test mismatch, which we alleviate by using a beam-search. We stop generation when all beams have produced an end-of-sequence token.

For both tracks, we train one model for each distinct pair of language and embedding architecture. We start by re-tokenizing the datasets using sentence piece with a vocabulary size of 15000. This is done in order to mitigate the effects of different

Team	en			es			fr			it			ru		
	Mv	SB	LB	Mv	SB	LB	Mv	SB	LB	Mv	SB	LB	Mv	SB	LB
Bl. SGNS	0.084	0.030	0.040	0.065	0.035	0.052	0.046	0.030	0.041	0.107	0.053	0.076	0.112	0.039	0.054
Bl. char	0.047	0.026	0.033	0.059	0.031	0.043	0.022	0.028	0.037	0.046	0.029	0.038	0.072	0.025	0.037
Bl. Electra	0.065	0.031	0.039				0.043	0.031	0.039				0.101	0.032	0.041
Locchi	0.049	0.022	0.027	0.038	0.020	0.026				0.071	0.008	0.012			
LingJing	−0.045	0.004	0.005	0.023	0.013	0.020	−0.113	0.003	0.005	−0.012	0.018	0.029	−0.010	0.011	0.014
BLCU-ICALL	**0.135**	0.031	0.040	**0.128**	0.039	0.056	0.042	0.027	0.037	**0.117**	**0.066**	**0.099**	**0.148**	0.048	0.065
IRB-NLP	0.094	**0.033**	0.042	0.093	**0.045**	**0.064**	0.056	0.028	0.033	0.077	0.010	0.015	0.080	0.027	0.036
RIGA	0.093	0.026	0.032	0.107	0.031	0.045	**0.075**	0.024	0.030	0.093	0.012	0.018	0.094	0.031	0.043
lukechan1231	0.071	0.022	0.027	0.068	0.025	0.036	0.054	0.021	0.026	0.101	0.037	0.054	0.109	0.029	0.040
Edinburgh	0.104	0.031	0.038	0.101	0.035	0.053	0.026	**0.029**	**0.038**	0.107	0.060	0.092	0.109	**0.049**	**0.072**
talent404	0.128	0.033	**0.043**												

Table 3: Participants' best scores on the Definition Modeling track. Highest participant scores per metric are displayed in bold font.

vocabulary sizes when training our Transformer baselines, and make the models overall easier to compare across different languages.

We set hyperparameters using a Bayesian Optimization procedure, with 100 hyperparameter configurations tested and 10 initial random samples. For the Reverse dictionary models, we tune the following hyper-parameters: learning rate, weight decay penalty, the β_1 and β_2 hyperparameters of the Adam optimizing algorithm, dropout rate, length of warmup, batch size,[11] number of heads in the multi-head attention layers, and number of stack layers. For the Definition Modeling systems, we also include a label smoothing parameter to tune. Models are trained over up to 100 epochs; training is stopped early if no improvement of at least 0.1% is observed during 5 epochs. In all cases, we decay the learning rate after the warmup following a half cosine wave, such that the learning rate reaches 0 at the end of the 100 epochs.

6 How whale did it go? Shared task results.

Scores attained by participants are shown in Tables 3 and 4. In Table 3, "Mv", "SB" and "LB" refer to Moverscore, Sense-BLEU and Lemma-BLEU respectively; in Table 4, each sub-table corresponds to a different architecure, and "rnk" refers to the cosine ranking metric (cf. Section 4).

In total, we received 159 valid submissions from 15 different users; out of which 11 teams produced a submission paper. 9 of these teams tackled the Definition Modeling, and 10 addressed the reverse dictionary track. Competition rankings are established by ranking each submission received, selecting for each participant the best performance on all metrics, and finally taking the average best rank. Some participants' submissions were faulty and could not be processed by the evaluation website scoring program.

Among the system descriptions we received, two focused solely on definition modeling. Kong et al. (2022, BLCU-ICALL) use a multitasking framework for definition modeling, based on a generation and a reconstruction objectives. Mukans et al. (2022, RIGA) focus on what are the effects of model size and duration of training on GRUs and LSTMs for definition modeling, and whether MoverScore corroborates human judgment.

Five submissions specifically focus on the reverse dictionary task. Bendahman et al. (2022, BL.research) compare the performances of MLP-based to LSTM-based networks for reverse dictionary. Li et al. (2022, LingJing) study pretraining objectives for the reverse dictionary track. Ardoiz et al. (2022, MMG) pay specific attention to how the not-so-satisfactory quality of the Spanish dataset impacts results on Spanish reverse dictionary. Cerniavski and Stymne (2022, Uppsala) study whether foreign language entries can improve the performance of the English reverse dictionary baseline model. Wang et al. (2022, 1cademy) introduce multiple technical tweaks for reverse dictionary, such as a dynamic weight averaging loss, language-specific tags and residual cutting.

[11] In practice, we first manually find the largest batch size that fits on our GPU, and then let the model select the number of batches it should accumulate gradient on.

Team	en			es			fr			it			ru		
	MSE	cos	rnk	MSE	cos	rnk	MSE	cos	rnk	MSE	cos	rnk	MSE	cos	rnk
Baseline	0.911	0.151	0.490	0.930	0.204	0.499	1.141	0.198	0.491	1.125	0.204	0.477	0.577	0.253	0.490
Locchi	0.875	0.204	0.394							1.087	0.274	0.386			
BL.research	0.895	0.166	0.312	0.910	0.252	0.253	1.107	0.212	0.314	1.111	0.246	0.247	0.566	0.298	0.290
LingJing	0.862	0.243	0.329	**0.858**	0.353	0.251	1.030	0.328	0.282	1.039	0.360	0.230	**0.528**	**0.424**	0.187
MMG				0.911	**0.403**	**0.167**									
chlrbgus321	**0.854**	0.248	0.319												
IRB-NLP	0.964	**0.260**	**0.231**	0.883	0.367	0.197	1.068	**0.342**	**0.193**	1.076	**0.380**	**0.165**	0.568	0.421	**0.150**
Edinburgh	0.864	0.241	0.326	0.860	0.347	0.271	**1.026**	0.312	0.302	**1.031**	0.374	0.197	0.538	0.383	0.247
the0ne	0.900	0.185	0.500												
JSI	0.909	0.156	0.499	0.913	0.223	0.495	1.122	0.216	0.498	1.196	−0.004	0.499	0.615	0.006	0.499
1cadamy	0.915	0.194	0.374	0.906	0.262	0.375	1.100	0.228	0.439	1.097	0.260	0.384	0.578	0.335	0.291

(a) SGNS Reverse Dictionary track results

Team	en			es			fr			it			ru		
	MSE	cos	rnk	MSE	cos	rnk	MSE	cos	rnk	MSE	cos	rnk	MSE	cos	rnk
Baseline	0.148	0.790	0.502	0.570	0.806	0.498	0.395	0.759	0.499	0.363	0.727	0.497	0.135	0.826	0.495
Locchi	**0.141**	**0.798**	0.483							0.355	0.734	0.478			
BL.research	0.143	0.795	0.450	0.510	0.824	0.412	0.366	0.770	0.428	0.359	0.728	0.417	0.132	0.830	0.410
LingJing	0.176	0.782	0.486	0.583	0.824	0.500	0.411	0.752	0.502	0.438	0.681	0.496	0.184	0.791	0.472
IRB-NLP	0.162	0.770	**0.419**	0.526	0.819	**0.403**	0.390	0.756	0.421	0.366	0.724	**0.383**	0.140	0.824	**0.357**
Edinburgh	0.143	0.795	0.500	**0.467**	**0.839**	0.424	**0.335**	**0.789**	0.428	**0.334**	**0.747**	0.428	**0.116**	**0.852**	0.389
the0ne	0.143	0.796	0.500												
1cadamy	0.168	0.792	0.478	0.557	0.820	0.410	0.391	0.769	**0.416**	0.364	0.739	0.438	0.156	0.836	0.377

(b) Char Reverse Dictionary track results

Team	en			fr			ru		
	MSE	cos	rnk	MSE	cos	rnk	MSE	cos	rnk
Baseline	1.413	0.843	0.498	1.153	0.856	0.498	0.874	0.721	0.491
Locchi	**1.301**	0.843	0.478						
BL.research	1.326	0.844	0.434	1.112	0.858	0.442	0.864	0.721	0.399
LingJing	1.509	0.846	0.478	1.271	0.859	0.478	**0.828**	0.734	0.420
IRB-NLP	1.685	0.828	**0.432**	1.339	0.847	**0.429**	0.911	0.724	**0.345**
Edinburgh	1.310	**0.847**	0.490	**1.066**	**0.862**	0.476	0.828	**0.735**	0.417
the0ne	1.340	0.846	0.500						

(c) ELECTRA Reverse Dictionary track results

Table 4: Participants' best scores on the Reverse Dictionary track. Highest participant scores per metric are displayed in bold font.

The last four submissions addressed both tracks. Chen and Zhao (2022, Edinburgh) propose to project embeddings and definitions on a shared representational space. Korenčić and Grubišić (2022, IRB-NLP) take inspiration from Noraset et al. (2017) to address definition modeling, and experiment with pooling strategies over Transformer embeddings for the reverse dictionary track. Tran et al. (2022, JSI) focus on comparing the effects of adding LSTM and BiLSTM layers on top of a Transformer model, as well as zero-shot cross-lingual generalization. Srivastava and Harsha Vardhan (2022, TLDR) propose two Transformer-based architectures for the two tracks, leveraging contrastive learning and unsupervised pretraining.

Looking at Tables 3 and 4, we see that the metrics we chose in section 4 are not always aligned. On the Definition Modeling track (Table 3), while the multitask framework of Kong et al. (2022, BLCU-ICALL) yields generally the most consistent performance, it is often outmatched in specific setups. For instance, BLEU-based metrics favor the shared

projection technique of Chen and Zhao (2022, Edinburgh) in Russian and French, while the pooling strategies of Korenčić and Grubišić (2022, IRB-NLP) appear especially effective on the Spanish dataset. As for the Reverse Dictionary track (Table 4), the strongest contender is generally the Edinburgh team, although the IRB-NLP team almost systematically produces the highest cosine ranking score. Interestingly, BLCU-ICALL, IRB-NLP and Edinburgh all rely on multi-task learning. Note however that the SGNS targets seem to depict a rather different picture, where the pretraining objectives of Li et al. (2022, LingJing) bring about some of the best results.

7 A deeper dive into our results

When looking at the competition results, two trends emerge. First, the baseline architectures from Section 5 remain quite competitive with solutions proposed by participants. Second, scores are generally unsatisfactory, especially in the definition modeling track: we do not see a clear divide between char embeddings and distributional semantic representations. The NLG metrics are, in absolute terms, low compared to modern NLP standards and results reported elsewhere on other definition modeling benchmarks. As for the reverse dictionary track, we see that across all submissions, at least a third of the test set is closer (in terms of cosine distance) to the production than the intended target.

Participants have suggested multiple reasons for these hardships. In particular, Ardoiz et al. (2022, MMG) highlight that the automated data compilation in DBnary (Sérasset, 2012) is of an unsatisfactory quality. Similar remarks can be made with respect to the embeddings, which are trained on rather small corpora. Other submissions such as Mukans et al. (2022, RIGA), Chen and Zhao (2022, Edinburgh), Korenčić and Grubišić (2022, IRB-NLP) highlight the limited applicability of mainstream NLG metrics, as we ourselves have discussed in Section 4.[12] One last remark is the limited size of our dataset, discussed by the Edinburgh and RIGA teams. All these remarks suggest avenues for future research: in particular, the release of the full dataset should alleviate some of the concerns with respect to dataset size. The MMG team also suggest some concrete preprocessing steps to handle some of the issues they identify in the proposed definitions.

[12] See also Mickus et al. (2021) for a discussion.

In terms of solutions explored, we can stress that teams have adopted a variety of strategies and architectures: systems used Transformer, RNN and CNN components, often leveraging or exploring multilingualism (Tran et al. 2022, JSI; Cerniavski and Stymne 2022, Uppsala; Wang et al. 2022, 1cademy; Bendahman et al. 2022, BL.research), multitasking, or multiple training objectives (Kong et al. 2022, BLCU-ICALL; 1cadamy; Korenčić and Grubišić 2022, IRB-NLP; Srivastava and Harsha Vardhan 2022, TLDR; Chen and Zhao 2022, Edinburgh). Multi-task training tends to yield varied yet competitive results for our data. No preponderant architecture emerges from the system descriptions; we note that multiple submissions based their work on other contextualized embedding architectures, trained from scratch on the CODWOE dataset (Wang et al. 2022, 1cademy; Li et al. 2022, LingJing). The comprehensive review of architectures by team 1cadamy suggests nonetheless that Transformers might be less suited to this shared task than recurrent models.

7.1 Manual analyses

As for manual evaluations, Kong et al. (2022, BLCU-ICALL) provide a thorough review of the errors produced by their model. Mukans et al. (2022, RIGA) provide some example outputs of their models, while Srivastava and Harsha Vardhan (2022, TLDR) and Wang et al. (2022, 1cademy) include ablation studies. The most thorough analysis, however, is that of Chen and Zhao (2022, Edinburgh), who provide both quantitative and qualitative (PCA-based) analyses across embedding architectures, languages, and trial dataset features. Korenčić and Grubišić (2022, IRB-NLP) provide an extremely well documented review of their systems performances, along multiple analyses of the embeddings proposed for the shared tasks, ranging from 2D down-projection visualizations to descriptive statistics of components. We refer the reader to the respective system papers for a more thorough review and focus here on a few promising approaches to summarize trends that emerge from these manual analyses.

Current metrics are not satisfactory. The IRB-NLP team highlight that the BLEU scores reported on the shared task are dramatically lower than what is generally expected in the literature; the Edinburgh team even shows that the S-BLEU scores obtained by non-sensical glosses such as "`, or .`"

can end up among the highest scores for some languages. The Reverse Dictionary metrics can also be sensitive to different aspects of the embeddings, as shown by the IRB-NLP team: this can lead to very different rankings of model productions, especially when comparing the cosine-based ranking metric to the cosine and MSE metrics. BLEU-based scores are also often sensitive to the length of the production, the target, or both, as shown by both the Edinburgh and the Riga teams.

Erroneous productions abound. Related to the previous remark, many Definition Modeling systems produce irrelevant or under-specified glosses, for which the proposed metrics are not satisfactory. For instance, the BLCU-ICALL report 52% irrelevant glosses and 23.5% under-specified glosses, from a manual evaluation of 200 productions. Other participating teams, such as RIGA or IRB-NLP, also display generated glosses with varying degress of semantic accuracy.

Embeddings contain more than semantics. The Edinburgh team highlights how different linguistic features retrieved from the trial dataset can significantly impact the scores they observe. They also highlight that char embeddings are separable by length, and that the Electra embeddings are clustered according to their frequency.

Not all setups are created equal. The Uppsala team report that Russian seems to be the most effective data source in their multilingual transfer experiments. The IRB-NLP team stresses that vector component distributions across languages and architectures as well as gloss length across languages can take very different values, and they also include 2D visualization suggesting the Electra embeddings tend to form neat cluster not observed for SGNS embeddings. Scores also vary quite a lot across setups (cf. Tables 3 and 4).

8 Conclusions and future perspectives

The CODWOE shared task was constructed so that participants' submissions would be likely to have linguistic significance. Yet, it is not trivial to tease apart the various factors that lead to the overall low results we observed. While the inadequacy of mainstream NLG metrics and the limitations of the dataset certainly play a role, they do not resolve the fundamental issue that we wished to investigate with CODWOE. Whether word embeddings and dictionaries contain the same information is not a solved research problem.

This has two immediate consequences: firstly, one can question the use of definition modeling as an evaluation tool for embeddings, as suggested by the seminal work of Noraset et al. (2017). The CODWOE shared task results indicate that the metrics currently used in the field are rife with caveats; in the controlled setup we have proposed here, participants rarely, if ever, found that character-based embeddings starkly contrasted with distributional semantic representations.

Second, one can question whether definition modeling and reverse dictionary are fit for building lexical resources for under-tooled languages: the crosslingual route proposed by Bear and Cook (2021) seems more practical than training models from scratch, even with relatively large datasets. Our embeddings were trained on corpora comparable in size to the 1B Words benchmark (Chelba et al., 2013): while modern text corpora are now several orders of magnitude larger, this dataset remained a landmark for several years. Our definitions were selected from DBnary (Sérasset, 2012), which focuses the largest Wiktionary projects.

Overall, the CODWOE shared task has been a success: we were able to show that the task at hand was far from trivial and we drew significant interest towards the issues addressed in the Definition Modeling and Reverse Dictionary literature. In future work, we plan to investigate better ways to perform NLG evaluation for the Definition Modeling task (in particular relying on human annotations) and we plan to focus on existing embeddings trained from very large corpora.

Acknowledgements

We thank the annotators who have helped us in putting an annotated dataset in a very short amount of time: Elena del Olmo Suárez, Hermès Martinez, Maria Copot, Nikolay Chepurnykh and Toma Gotkova, as well as Cyril Pestel for his help with data hosting.

This work was also supported by a public grant overseen by the French National Research Agency (ANR) as part of the "Investissements d'Avenir" program: Idex *Lorraine Université d'Excellence* (reference: ANR-15-IDEX-0004).

References

Alfonso Ardoiz, Miguel Ortega-Martín, Óscar García-Sierra, Jorge Álvarez, Ignacio Arranz, and Adrián Alonso. 2022. MMG at SemEval-2022 Task 1: A reverse dictionary approach based on a review of the dataset from a lexicographic perspective.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Diego Bear and Paul Cook. 2021. Cross-lingual wolastoqey-English definition modelling. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 138–146, Held Online. INCOMA Ltd.

Nihed Bendahman, Julien Breton, Lina Nicolaieff, Mokhtar Boumedyen Billami, Christophe Bortolaso, and Youssef Miloudi. 2022. BL.Research at SemEval-2022 Task 1: Deep networks for reverse dictionary using embeddings and lstm autoencoders.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Slaven Bila, Wataru Watanabe, Taiichi Hashimoto, Takenobu Tokunaga, and Hozumi Tanaka. 2004. Dictionary search based on the target word description. In *Proceedings of the 10th Annual Meeting of the Association for Natural Language Processing (ANLP 2004)*.

Gemma Boleda. 2020. Distributional semantics and linguistic theory. *Annual Review of Linguistics*, 6(1):213–234.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Hiram Calvo, Oscar Méndez, and Marco A. Moreno-Armendáriz. 2016. Integrated concept blending with vector space models. *Comput. Speech Lang.*, 40(C):79–96.

Rafal Cerniavski and Sara Stymne. 2022. Uppsala university at SemEval-2022 Task 1: Multilingualism in reverse dictionaries: Can foreign entries enhance an english reverse dictionary?

Ciprian Chelba, Tomás Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Phillipp Koehn. 2013. One billion word benchmark for measuring progress in statistical language modeling. *CoRR*, abs/1312.3005.

Pinzhen Chen and Zheng Zhao. 2022. Edinburgh at SemEval-2022 Task 1: Jointly fishing for word embeddings and definitions.

Kevin Clark, Minh-Thang Luong, Quoc Le, and Christopher D. Manning. 2020. Pre-training transformers as energy-based cloze models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 285–294, Online. Association for Computational Linguistics.

Isabelle Dautriche, Kyle Mahowald, Edward Gibson, and Steven T. Piantadosi. 2017. Wordform similarity increases with semantic similarity: An analysis of 100 languages. *Cognitive Science*, 41(8):2149–2169.

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.

Dominique Dutoit and Pierre Nugues. 2002. A lexical database and an algorithm to find words from definitions. In *Proceedings of the 15th European Conference on Artificial Intelligence*, ECAI'02, page 450–454, NLD. IOS Press.

Ilknur Durgar El Khalout and Kemal Oflazer. 2004. Use of wordnet for retrieving words from their meanings. In *Proceedings of the Second Global Wordnet Conference (GWC 2004)*, pages 118–123.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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 55–65, Hong Kong, China. Association for Computational Linguistics.

Artyom Gadetsky, Ilya Yakubovskiy, and Dmitry Vetrov. 2018. Conditional generators of words definitions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 266–271, Melbourne, Australia. Association for Computational Linguistics.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information*

Processing Systems, volume 27. Curran Associates, Inc.

E. Dario Gutiérrez, Roger Levy, and Benjamin Bergen. 2016. Finding non-arbitrary form-meaning systematicity using string-metric learning for kernel regression. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2379–2388, Berlin, Germany. Association for Computational Linguistics.

Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

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.

Arman Kabiri and Paul Cook. 2020. Evaluating a multisense definition generation model for multiple languages. In *Text, Speech, and Dialogue*, pages 153–161, Cham. Springer International Publishing.

Cunliang Kong, Yujie Wang, Ruining Chong, Liner Yang, Hengyuan Zhang, Erhong Yang, and Yaping Huang. 2022. BLCU-ICALL at SemEval-2022 Task 1: Cross-attention multitasking framework for definition modeling.

Damir Korenčić and Ivan Grubišić. 2022. IRB-NLP at SemEval-2022 Task 1: Exploring the relationship between words and their semantic representations.

Andrei Kutuzov. 2017. Arbitrariness of linguistic sign questioned: Correlation between word form and meaning in russian.

Alessandro Lenci. 2018. Distributional models of word meaning. *Annual Review of Linguistics*, 4(1):151–171.

Bin Li, Yixuan Weng, Fei Xia, Shizhu He, Bin Sun, and Shutao Li. 2022. Lingjing at SemEval-2022 Task 1: Multi-task self-supervised pre-training for multilingual reverse dictionary.

Yinqiao Li, Chi Hu, Yuhao Zhang, Nuo Xu, Yufan Jiang, Tong Xiao, Jingbo Zhu, Tongran Liu, and Changliang Li. 2020. Learning architectures from an extended search space for language modeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6629–6639, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Arman Malekzadeh, Amin Gheibi, and Ali Mohades. 2021. PREDICT: persian reverse dictionary. *CoRR*, abs/2105.00309.

Oscar Méndez, Hiram Calvo, and Marco A. Moreno-Armendáriz. 2013. A reverse dictionary based on semantic analysis using wordnet. In *Advances in Artificial Intelligence and Its Applications*, pages 275–285, Berlin, Heidelberg. Springer Berlin Heidelberg.

Timothee Mickus, Mathieu Constant, and Denis Paperno. 2020a. Génération automatique de définitions pour le français (definition modeling in French). In *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 2 : Traitement Automatique des Langues Naturelles*, pages 66–80, Nancy, France. ATALA et AFCP.

Timothee Mickus, Mathieu Constant, and Denis Paperno. 2021. About neural networks and writing definitions. *Dictionaries: Journal of the Dictionary Society of North America*, 42(2).

Timothee Mickus, Denis Paperno, Mathieu Constant, and Kees van Deemter. 2020b. What do you mean, BERT? In *Proceedings of the Society for Computation in Linguistics 2020*, pages 279–290, New York, New York. Association for Computational Linguistics.

Timothee Mickus, Denis Paperno, and Matthieu Constant. 2019. Mark my word: A sequence-to-sequence approach to definition modeling. In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 1–11, Turku, Finland. Linköping University Electronic Press.

Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Eduards Mukans, Gus Strazds, and Guntis Barzdins. 2022. RIGA at SemEval-2022 Task 1: Scaling recurrent neural networks for CODWOE dictionary modeling.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3259–3266. AAAI Press.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia,

Pennsylvania, USA. Association for Computational Linguistics.

Tiago Pimentel, Arya D. McCarthy, Damian Blasi, Brian Roark, and Ryan Cotterell. 2019. Meaning to form: Measuring systematicity as information. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1751–1764, Florence, Italy. Association for Computational Linguistics.

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.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

Ferdinand de Saussure. 1916. *Cours de linguistique générale*, 1995 edition. Payot & Rivage, Paris.

Gilles Sérasset. 2012. Dbnary: Wiktionary as a LMF based multilingual RDF network. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2466–2472, Istanbul, Turkey. European Language Resources Association (ELRA).

Ryan Shaw, Anindya Datta, Debra VanderMeer, and Kaushik Dutta. 2013. Building a scalable database-driven reverse dictionary. *IEEE Transactions on Knowledge and Data Engineering*, 25(3):528–540.

Bushra Siddique and Mirza Mohd Sufyan Beg. 2019. A review of reverse dictionary: Finding words from concept description. In *Next Generation Computing Technologies on Computational Intelligence*, pages 128–139, Singapore. Springer Singapore.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Aditya Srivastava and Vemulapati Harsha Vardhan. 2022. TLDR at SemEval-2022 Task 1: Using transformers to learn dictionaries and representations.

Sushrut Thorat and Varad Choudhari. 2016. Implementing a reverse dictionary, based on word definitions, using a node-graph architecture. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2797–2806, Osaka, Japan. The COLING 2016 Organizing Committee.

William Timkey and Marten van Schijndel. 2021. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4527–4546, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thi Hong Hanh Tran, Matej Martinc, Matthew Purver, and Senja Pollak. 2022. JSI at SemEval-2022 Task 1: CODWOE - reverse dictionary: Monolingual and cross-lingual approaches.

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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Zhiyong Wang, Ge Zhang, and Nineli Lashkarashvili. 2022. 1cademy at SemEval-2022 Task 1: Investigating the effectiveness of multilingual, multitask, and language-agnostic tricks for the reverse dictionary task.

Hang Yan, Xiaonan Li, Xipeng Qiu, and Bocao Deng. 2020. BERT for monolingual and cross-lingual reverse dictionary. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4329–4338, Online. Association for Computational Linguistics.

Liner Yang, Cunliang Kong, Yun Chen, Yang Liu, Qinan Fan, and Erhong Yang. 2019. Incorporating sememes into chinese definition modeling. ArXiv preprint : 1905.06512.

David Yenicelik, Florian Schmidt, and Yannic Kilcher. 2020. How does BERT capture semantics? a closer look at polysemous words. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 156–162, Online. Association for Computational Linguistics.

Fabio Massimo Zanzotto, Ioannis Korkontzelos, Francesca Fallucchi, and Suresh Manandhar. 2010. Estimating linear models for compositional distributional semantics. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1263–1271, Beijing, China. Coling 2010 Organizing Committee.

Haitong Zhang, Yongping Du, Jiaxin Sun, and Qingxiao Li. 2020a. Improving interpretability of word embeddings by generating definition and usage. *Expert Systems with Applications*, 160:113633.

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

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 *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 563–578, Hong Kong, China. Association for Computational Linguistics.