Benchmarking Meta-embeddings: What Works and What Does Not

Iker García-Ferrero **Rodrigo Agerri** **German Rigau**
HiTZ Basque Center for Language Technologies - Ixa NLP Group
University of the Basque Country UPV/EHU
{ iker.garciaf, rodrigo.agerri, german.rigau }@ehu.eus

Abstract

In the last few years, several methods have been proposed to build meta-embeddings. The general aim was to obtain new representations integrating complementary knowledge from different source pre-trained embeddings thereby improving their overall quality. However, previous meta-embeddings have been evaluated using a variety of methods and datasets, which makes it difficult to draw meaningful conclusions regarding the merits of each approach. In this paper we propose a unified common framework, including both intrinsic and extrinsic tasks, for a fair and objective meta-embeddings evaluation. Furthermore, we present a new method to generate meta-embeddings, outperforming previous work on a large number of intrinsic evaluation benchmarks. Our evaluation framework also allows us to conclude that previous extrinsic evaluations of meta-embeddings have been overestimated.

1 Introduction

Word embeddings successfully capture lexical semantic information about words based on co-occurrence patterns extracted from large corpora (Mikolov et al., 2013a; Pennington et al., 2014; Mikolov et al., 2018) or knowledge bases (Bordes et al., 2011), with excellent results on several tasks, including word similarity (Collobert and Weston, 2008; Turian et al., 2010; Socher et al., 2011), Semantic Textual Similarity (Shao, 2017), or more recently, unsupervised machine translation (Artetxe et al., 2019), inferring representations for rare words (Schick and Schütze, 2020), unsupervised word alignment (Jalili Sabet et al., 2020) or knowledge base probes (Dufter et al., 2021). In these tasks, word embeddings perform similarly or better than transformer-based language models such as BERT (Devlin et al., 2019), while requiring a comparatively tiny amount of resources for training and inference.

Following the hypothesis that different knowledge sources may contain complementary semantic information (Goikoetxea et al., 2016), meta-embeddings (Yin and Schütze, 2016) aim to obtain an ensemble of distinct word embeddings each trained using different methods and resources to produce a word representation with an improved overall quality.

The main challenge when generating meta-embeddings is preserving the information encoded in the source embeddings and many different methods have been proposed to deal with the task. Concatenation (Goikoetxea et al., 2016) and averaging (Coates and Bollegala, 2018) are two very strong baselines, but much complex methods based on linear transformations and supervised neural models have also been proposed (Bollegala et al., 2018; Bollegala and Bao, 2018; Yin and Schütze, 2016).

When it comes to evaluating meta-embeddings, there is no consensus on either evaluation tasks or methodology. Meta-embeddings are evaluated in a wide range of tasks (Schnabel et al., 2015; Bakarov, 2018), ranging from intrinsic (i.e. word similarity, word analogy) to extrinsic tasks such as short text classification (Bollegala and Bao, 2018; Bollegala et al., 2018), common-sense stories (Speer et al., 2017), Named Entity Recognition (O'Neill and Bollegala, 2020) or Semantic Textual Similarity (García-Ferrero et al., 2020). Furthermore, different evaluation methodologies have been applied. For example, Yin and Schütze (2016) discard the words in the datasets which are not represented in the meta-embedding model, while Speer and Lowry-Duda (2017) use various strategies to minimize the number of out-of-vocabulary (OOV) words. To make things more complicated, previous meta-embeddings approaches require some ad-hoc pre-processing to tune multiple filtering criteria and parameters according to the source embeddings used (Bollegala et al., 2018; Bollegala and Bao, 2018; Yin and Schütze, 2016), which has a signifi-

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 3957–3972
November 7–11, 2021.

cant effect on the final evaluation results. Summarizing, this lack of consistency in evaluation tasks, methodologies and ad-hoc hyper-parameter tuning makes it very hard to objectively compare the proposed methods. Thus, to the best of our knowledge, and despite the existence of multiple works addressing this task, a unified and comprehensive evaluation of meta-embeddings has not been yet carried out. In fact, the lack of such unified and comprehensive evaluation framework has arguably caused erroneous assumptions and an overestimation in the performance of meta-embeddings for extrinsic tasks.

An additional issue is that most previous work has focused on combining word embeddings generated from similar sources and algorithms. For instance, combining Word2vec CBOW (Mikolov et al., 2013a) with GloVe (Pennington et al., 2014) embeddings. We empirically show that, since these embeddings encode very similar knowledge, combining them does not produce a significant gain. Instead, the best meta-embeddings are obtained by combining embeddings trained with different algorithms and resources. For example, by leveraging vectors induced from text corpora together with other embeddings obtained from knowledge bases.

In this paper we present a *new method to generate meta-embeddings* that outperform previous approaches on a *large number of intrinsic benchmarks*. Other contributions include:

1. We empirically demonstrate that our method generates better meta-embeddings thanks to *decreasing the information loss* during the embedding combination. Our approach does not rely on hyper-parameter tuning.

2. We generate meta-embeddings using a wide range of source embeddings trained with very different algorithms and resources. Our experiments show that *the best meta-embeddings are obtained when combining embeddings that encode complementary knowledge*.

3. A *unified and comprehensive benchmarking framework* to facilitate a fair and objective evaluation of embeddings in both intrinsic and extrinsic settings.

4. We report the *largest meta-embedding extrinsic evaluation* performed so far showing that meta-embedding performance in these tasks has been overestimated by previous work.

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 focuses on the evaluation frameworks used by previous works and presents our own proposal. In Section 4 we describe our approach for creating meta-embeddings, with Section 5 describing the source word embeddings explored and reporting our experimental results in Section 6. Finally, Section 7 presents some concluding remarks and our future work. Our code and meta-embeddings are publicly available[1].

2 Related work

Previous research has shown that word embeddings created using different methods and resources present significant variations in quality. For instance, Hill et al. (2014) show that word embeddings trained from monolingual or bilingual corpora capture different nearest neighbours.

The term meta-embedding was coined by Yin and Schütze (2016). They showed how to combine five different pre-trained word embeddings using a small neural network for improving the accuracy of cross-domain part-of-speech (POS) tagging. Following this, Bollegala et al. (2018) propose an unsupervised locally linear method for learning meta-embeddings from a given set of pre-trained source embeddings while Bollegala and Bao (2018) apply three types of autoencoders for the purpose of learning meta-embeddings.

Although word embeddings are mainly constructed by exploiting information from text corpora only (Mikolov et al., 2013a; Pennington et al., 2014; Mikolov et al., 2018), some approaches also tried different methods to integrate the knowledge encoded in lexical resources such as WordNet (Halawi et al., 2012; Bollegala et al., 2016; Goikoetxea et al., 2016), PPDB (Faruqui et al., 2015) or ConceptNet (Speer et al., 2017). Goikoetxea et al. (2016) show that simply concatenating word embeddings derived from text and WordNet outperform alternative methods such as retrofitting (Faruqui et al., 2015) at the cost of increasing the dimensionality of the meta-embeddings. Coates and Bollegala (2018) prove that averaging is in some cases better than concatenation, with the additional benefit of a reduced dimensionality. The most popular approach to address the dimensionality problem is to apply dimensionality reduction algorithms

[1] https://github.com/ikergarcia1996/MetaVec

Paper	Intrinsic Tasks	Extrinsic Tasks
(Kiela et al., 2018)		SST, SNLI, Image Caption (1)
(He et al., 2020)		SST2, SNLI, NER (1), POS(1), Semcor
(Bollegala and Bao, 2018)	Sim. (4), An. (3), Relation Classification (1)	Short Text Classification (4), Psycholinguistic Score Prediction (2)
(O'Neill and Bollegala, 2020)	Sim. (6), An. (3)	POS (1), NER (1), Sentiment Analysis(1)
(Jawanpuria et al., 2020)	Sim. (6), An. (2)	
(Doval et al., 2018)	Sim. (4), Bilingual dictionary induction (4), hypernym discovery (1)	
(Bollegala et al., 2018)	Sim. (6), An. (2), Relation Classification (1)	Short-text classification (2)
(Coates and Bollegala, 2018)	Sim. (5), An. (1)	
(Yin and Schütze, 2016)	Sim. (5), An. (1)	POS (1)
(Li et al., 2020)		MT (3), Text Classification (5)
(Chen et al., 2020)	Sim. (5)	SNLI (1)
(Goikoetxea et al., 2016)	Sim. (4)	
(Speer et al., 2017)	Sim. (5). SAT An. (1)	Common-Sense Stories (1)
(García-Ferrero et al., 2020)	Sim. (14)	STS (1), POS (1)
This work	Sim. (7), An. (3), Categorization (4)	CoLA, SST-2, MRPC, STS-B, QQP, MNLI, QNLI, RTE , WNLI, AX

Table 1: Evaluation tasks used in previous works.

	w OOV	w/o OOV
Default	82.7	82.7
Clean dataset	69.8	74.3
Lowercase embedding	**39.5**	80.7
Trim vocabulary	40.4	**84.1**

Table 2: FastText embeddings accuracy in the Google Analogy dataset using different pre-processing approaches.

such as SVD (Yin and Schütze, 2016), PCA (Ghannay et al., 2016) or DRA (Raunak, 2017). In this line of work, Numberbatch (Speer et al., 2017) claims to be the best meta-embedding model so far, by combining knowledge from a variety of embeddings obtained from different corpora and knowledge bases such as ConceptNet.

Methods such as MUSE (Lample et al., 2018) and VecMap (Artetxe et al., 2018) project embeddings of two different languages to a shared common space by means of a bilingual dictionary (Mikolov et al., 2013b). This requires minimal bilingual supervision while still leveraging large amounts of monolingual corpora with very competitive results (Artetxe et al., 2016, 2018). These techniques are used by Doval et al. (2018); García-Ferrero et al. (2020); Jawanpuria et al. (2020); He et al. (2020) to generate meta-embeddings. This usually involves mapping all the source embeddings to a common vector space followed by averaging. We extend this idea by proposing a multiple step algorithm that: (i) normalizes the source embeddings; (ii) maps them to the same vector space; (iii) handles the OOV words; and (iv), generates the final meta-embedding. An ablation study confirms that these steps increase the performance of the generated meta-embeddings in both intrinsic and extrinsic tasks.

Another recent research line tries to dynamically generate meta-embeddings for specific tasks (He et al., 2020; Kiela et al., 2018; O'Neill and Bollegala, 2020). These methods extend already existing algorithms to generate meta-embeddings by learning task specific weights. Instead, the focus of our research is to generate the best general purpose meta-embedding that can be applied to any task.

3 Evaluation Framework

As it has been earlier mentioned, several methods to generate meta-embeddings have been previously proposed and evaluated on many different benchmarks, as shown by Table 1. Moreover, add-hoc decisions (not always explicitly mentioned) to evaluate the embeddings caused large variations in the results. Let us consider, for example, the problem of out-of-vocabulary (OOV) words.

Two popular techniques are used to address OOV words. Table 2 shows the accuracy of FastText embeddings [2] in the Google Analogy dataset using the two approaches. The first one uses the average of all the embeddings as a representation for unknown words (With OOV). The second approach simply removes from the dataset the examples containing unknown words (Without OOV). Additionally, the dataset is usually pre-processed. A common approach lowercase all the words and removes non

[2]Trained in Common Crawl corpus with 600B tokens.

English characters (Clean dataset) to reduce the number of unknown words. The words in the embedding can also be lowercased (Lowercase embeddings). Another popular practice to evaluate analogy consist of trimming the vocabulary of the embedding to the k most popular words. As an example, trimming the vocabulary to the 100,000 most popular English words also speeds up the computations (Trim vocabulary). These changes in the pre-processing of the very same embeddings cause the results to vary from 39.5% accuracy to 84.1%. Obviously, without a common evaluation framework the comparison between the different embeddings and meta-embeddings cannot be objectively done.

This lack of evaluation consistency led us to propose a unified evaluation framework that encompasses a wide range of tasks and datasets to evaluate meta-embeddings. In order to make the evaluation as simple and unified as possible we chose two already existing out of the box frameworks:

Word embeddings benchmarks[3] (Jastrzebski et al., 2017) provides scripts for evaluating word embeddings in *three intrinsic* evaluation tasks: (i) **Word similarity** (WS353 (Finkelstein et al., 2001), MTurk (Halawi et al., 2012), RG65 (Rubenstein and Goodenough, 1965), RW (Pilehvar et al., 2018), SimLex999 (Hill et al., 2015), MEN (Bruni et al., 2014)); (ii) **Word analogy** (Google Analogy (Mikolov et al., 2013a), MSR Analogy (Mikolov et al., 2013c), SemEval2012 (Jurgens et al., 2012)) and, (iii) **Word categorization** (AP (Almuhareb and Poesio, 2005), BLESS (Baroni and Lenci, 2011), Battig (Battig and Montague, 1969), ESSLI (McRae et al., 2005)). We use the provided script for evaluating embeddings on all the tasks without lowercasing them.

It should be taken into account that, for Word analogy, smaller vocabularies usually obtain better results. This particularly hurts the performance of those meta-embeddings that were generated using many source embeddings resulting in a meta-embedding with a vocabulary of more than 4 million words. Thus, in order to ensure a fair evaluation regardless of the number of words in the vocabulary, we trim the vocabulary of all the embeddings and meta-embeddings to the 200,000 most popular English words according to the Google's Trillion Word Corpus [4].

Jiant[5] provides a framework for extrinsic evaluation of word representations using GLUE (Wang et al., 2019b) and SuperGLUE (Wang et al., 2019a). We use the same bag-of-words configuration used in the GLUE leaderboard for the Cbow baseline [6] and we evaluate the embeddings in all GLUE tasks (CoLa (Warstadt et al., 2019) , SST-2 (Socher et al., 2013), MRPC (Dolan and Brockett, 2005) , STS-B (Cer et al., 2017), QQP [7], MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016; Wang et al., 2019b), RTE (Dagan et al., 2006; Bar Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009), WNLI (Levesque et al., 2011), AX (Wang et al., 2019b)).

4 Our Method

Our meta-embedding generation approach consists of two main steps: (i) pre-processing of the source embeddings and (ii) generation of the meta-embedding by averaging. Our method can combine any number of word embeddings as long as there is some common vocabulary shared between them. The resulting meta-embedding vocabulary will be the union of the vocabularies of the source word embeddings used.

4.1 Word embeddings pre-processing

Word embeddings generated with different sources or techniques can result in very different vectors spaces and vocabularies. Before aligning the vector spaces an harmonization pre-processing step is needed. Thus, we translate, scale, rotate and match the vocabularies of the source embeddings.

1) Mean Centering and scaling: Following (Artetxe et al., 2018), we first normalize the length of the source embeddings. We mean center each dimension, and we normalize them again by length. This translates all the source embeddings to the origin and scales them to have the same length.

2) Aligning the vector spaces: We align the vector spaces of the source embeddings using VecMap (Artetxe et al., 2016). VecMap learns word embedding mappings using an orthogonal

[3] https://github.com/kudkudak/word-embeddings-benchmarks

[4] https://books.google.com/ngrams/info

[5] https://github.com/nyu-mll/jiant-v1-legacy

[6] https://github.com/nyu-mll/jiant-v1-legacy/blob/master/jiant/config/superglue_bow.conf

[7] https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

transformation. Orthogonality allows monolingual invariance during the mapping, preserving vector dot products between word vectors. Monolingual invariance ensures that no information is lost during the mapping step, which is desirable for our aim of generating meta-embeddings. In our experiments we align the source embedding by projecting them to the vector space of one particular source embeddings involved in the construction of the meta-embeddings.

3) OOV generation: Different word embeddings have different vocabularies. When combining two word embeddings we can distinguish two sets of words. Those for which we have a representation in both embeddings and those for which one of the embeddings has no representation. We call the latter "OOV words". We unify the vocabulary of the source embeddings by creating new approximate representations for the OOV words.

The process is as follows. Given two source embeddings E1 and E2 where for a word W only E1 has a representation, we generate a new approximation for the OOV word in E2 by revising the most similar words from the common vocabulary of E1 and E2. First, using the cosine similarity as distance metric, we select the k (ranging from 2 to 50) nearest neighbours of the word W in E1 that also appear in the common vocabulary with E2.[8] For each k, we calculate k candidate representations of the OOV word in E2 and E1 as a weighted average of the selected k nearest neighbours in their corresponding spaces. We use the cosine similarity from the nearest neighbors in E1 to W as weights. Finally, the selected representation of the OOV word in E2 is the one corresponding to the closest candidate to W in E1.

4.2 Meta-embedding generation

We combine the harmonized source embeddings by averaging them. In our experiments we demonstrate that, thanks to the pre-processing steps described above, averaging source embeddings effectively combines multiple source embeddings resulting in representations as good as the ones generated by concatenation without increasing their dimensionality.

[8]For computation efficiency we limit the maximum k to 50. In our experiments the optimal k is usually smaller than 20.

5 Word embeddings

This section describes the source word embeddings used to generate our meta-embeddings. We choose these pre-trained embeddings for two main reasons. They have been trained using very diverse algorithms and resources, and they obtain good performance on our evaluation framework when tested individually. That is, they may encode high quality complementary knowledge.

Using Large text corpora, **Word2Vec** (W2V) (Mikolov et al., 2013a) embeddings from Google News (100 billion words). A **GloVe** (GV) (Pennington et al., 2014) model the Common Crawl vectors (640 billion words). As recommended by the authors, we apply a $l2$ normalization to its variables. And the **FastText** (FT) (Mikolov et al., 2018) embeddings from Common Crawl (600 billion words).

Using WordNet (Miller, 1992), **RWSGwn** (UKB) (Goikoetxea et al., 2015) combines random walks over WordNet with the skip-gram model. We have used the vectors trained using WordNet3.0 plus gloss relations. **JOINTChyb** (J) (Goikoetxea et al., 2018) combines Random Walks over multilingual WordNets and bilingual corpora as input for a modified skip-gram model that forces equivalent terms in different languages to come closer during training. We used the English-Spanish bilingual embeddings publicly available.

Using the Paraphrase Database (PPDB) (Ganitkevitch et al., 2013), **Attract Repel** (AR) (Mrkšić et al., 2017) improves word embeddings by injecting synonymy and antonym constraints extracted from monolingual and cross-lingual lexical resources. We used the English vocabulary from the four-lingual (English, German, Italian, Russian) vector space. **Paragram** (P) (Wieting et al., 2015) are pre-trained word vectors learned using word paraphrase pairs from PPDB using a modification of the skip-gram objective function. The hyper parameters were tuned using the wordsim-353 dataset. The word embeddings of the default model are initialized with Glove word vectors.

Using ConceptNet; **Numberbatch** (N) (Speer et al., 2017) combines knowledge encoded in ConceptNet, Word2vec, GloVe and OpenSubtitles 2016 using concatenation, dimensionality reduction and a variation of retrofitting. Numberbatch version 19.08 is used.

We also tested other embeddings such as ExtVec (Komninos and Manandhar, 2016), LexSub (Arora et al., 2020) or LexVec (Salle et al., 2016) but

Embedding	AVG	C	WS	A
FT	67.8	71.4	73.6	**58.5**
GV	64.7	69.9	70.3	54.0
W2V	59.1	67.9	65.6	43.9
J	52.2	70.0	65.2	21.4
UKB	46.6	67.9	61.8	10.2
P	58.5	66.5	70.2	38.9
AR	48.5	59.7	63.6	22.2
N	**68.1**	**73.6**	**75.2**	55.4

Table 3: Source embedding intrinsic evaluation results.

	Text	WN	PPDB	CN
Text	67.9	66.3	68.5	69.1
WN	66.3	50.9	62.5	65.4
PPDB	68.5	62.5	60.2	67.8
CN	69.1	65.4	67.8	-

Table 4: Comparison of the average performance in the intrinsic evaluation tasks for meta-embeddings generated using pairs of embeddings that encode knowledge form the same or different sources. WN stands for WordNet and CN for ConceptNet.

showed no significant improvements over the chosen ones.

6 Experiments

We evaluate all the word embeddings in a wide range of intrinsic and extrinsic evaluation tasks which composed the evaluation framwework described in Section 3.

6.1 Intrinsic evaluation results

First we evaluate the source embeddings that we will later use for meta-embedding generation. Table 3 shows the averaged results of the Categorization (C), Word Similarity (WS) and Analogy (A) datasets. We report the average cluster purity score of the Categorization datasets, the average Spearman correlation in the WS datasets, and the average score[9] in the Word Analogy datasets. The results shows that FastText achieve the best performance on the Analogy datasets and Numberbatch on Categorization and Word Similarity. As expected, on average Numberbatch obtains the best results on the intrinsic evaluations tasks.

We start generating meta-embeddings with our proposed method combining pairs of source embeddings. Table 4 shows the average score in the intrinsic evaluation benchmark of different pairs of source embeddings. For each source class type (Text Corpora, WordNet, PPDB and ConceptNet), we combine the best embeddings of each class with the best embeddings of the other classes. Within the same class we combine the first and second best embeddings.

The results show that, instead of using embeddings based on the same information type, combining embeddings of different classes obtains most of the time better results. That is, two embeddings generated using similar sources do not contain complementary knowledge, and its combination does not result in better performance. In our experiments, the best results are achieved when combining source embeddings generated using very different resources, such as text and knowledge bases. These combinations produce a meta-embedding that encodes the complementary knowledge of the source embeddings resulting in an improved performance. Also note that the meta-embedding combining text (FT) and PPDB (P), and also text (FT) with ConceptNet (N) outperforms the results of Numberbach (N) alone.

We generate our best meta-embeddings combining the best source embeddings created using large text corpora (FT), WordNet (J), PPDB (P) and ConceptNet (Numberbatch) (hereinafter **FJNP**). This combination maximizes the complementary knowledge encoded in the meta-embedding. We compare our method with 3 baselines using the same source embeddings: (i) **Concatenation:** (CONC+) Concatenation is a very strong baseline in meta-embedding generation. It allows combining multiple embeddings without any information loss. However, this comes at a high cost, as the meta-embedding dimensionaly is increased dramatically. We standardize the source embeddings using the approach described in Section 4.1. (ii) **AutoEncoders** (Bollegala and Bao, 2018): Autoencoders are an unsupervised learning method that first compress the input in a space of latent variables and then reconstructs the input based on the information encoded in these latent variables. It aims to learn meta-embeddings by reconstructing multiple source embeddings. This method comes in three flavours, DAEME, CAEME and AAEME. We used the last one because it obtains the best results. We applied the default parameters and enabled the option to generate OOV word representations. (iii) **Locally Linear Meta-Embedding**

[9]We calculate the Spearman Correlation for the SemEval2012 dataset and accuracy for GoogleAnalogy and MSR

FJNP	AVG	C	WS	A
CONC+	70.1	71.7	**78.5**	**60.1**
LLE	52.4	60.8	68.1	28.3
AAEME	67.6	71.2	75.0	56.6
Our Method	**70.6**	**73.5**	78.4	59.9

Table 5: Comparison of our meta-embedding method, baselines and prior work in the intrinsic evaluation.

Learning (LLE) (Bollegala et al., 2018): This approach which consists of two steps. In the reconstruction step the embeddings of each word are represented by the linear weighted combination of the embeddings of its nearest neighbours. In the projection step the meta-embedding of each word is computed such that the nearest neighbours in the source embedding spaces are embedded closely to each other in the meta-embedding space. We tested this method with the same parameters used in the original paper. Note that the code provided by the authors generates meta-embeddings using the intersection of the vocabulary of the source embeddings. This results in a small vocabulary that severely hurts its performance in some tasks.

Table 5 reports the results for our method and the baselines. The overall performance of our method is slightly better than concatenation (improved with our standardization method), mostly due to the good results in Categorization. In any case, the most important point here is to notice that our method, unlike concatenation (CONC+), does not increase the final dimensionality of the meta-embeddings. Furthermore, our technique clearly outperforms the meta-embeddings generated by Autoencodding and LLE and all the embeddings listed in Table 3 including Numberbatch, which is a meta-embedding. To the best of our knowledge, these are the best results published using these intrinsic benchmarks.

6.2 Extrinsic evaluation results

We compare our meta-embeddings with the same source embeddings and baselines used in the intrinsic evaluation (subsection 6.1). We test the same combination of embeddings that provides the best results in the intrinsic evaluation (FJNP). For brevity we report the GLUE Score calculated as proposed by the authors (Wang et al., 2019b). We are aware that, for the GLUE benchmark, (static) word embeddings are outperformed by contextual representations such as those obtained by BERT (Devlin et al., 2019). Thus, word embeddings may be better suited for other tasks such as unsupervised machine translation (Artetxe et al., 2019), inferring high-quality embeddings for rare words (Schick and Schütze, 2020), unsupervised word alignment (Jalili Sabet et al., 2020) or knowledge base queries (Dufter et al., 2021). However, we can use the GLUE benchmark as part of an objective and unified framework to evaluate word embeddings. In this sense, future research can also use exactly the same setting and methodology to evaluate new word embeddings and meta-embeddings.

Table 6 presents the results of the extrinsic evaluation. Interestingly, FastText achieves the best results, outperforming every single meta-embedding in every task. In fact, Numberbatch and AAEME fail on the extrinsic evaluation achieving very low results compared with the source word embeddings.

Previous research in meta-embedding generation has limited the extrinsic evaluation to very few tasks that are formulated closely to the intrinsic evaluation such as short text classification (Bollegala and Bao, 2018; Bollegala et al., 2018) or common-sense stories (Speer et al., 2017). Other approaches combine meta-embeddings with contextual representations with the aim of achieving SOTA results for tasks such as STS or POS tagging (García-Ferrero et al., 2020). While those previous works assume that meta-embeddings might be helpful for such extrinsic evaluation tasks, our results show that when evaluating on ten challenging tasks, FastText is indeed a very strong baseline that is not improved by any meta-embedding proposed up to date. These results suggest that meta-embeddings generated using complementary knowledge from WordNet, ConceptNet or PPDB help to improve performance for intrinsic tasks, but that this is not the case for extrinsic evaluations using GLUE.

6.3 Ablation study

We perform an ablation study to determine which steps of our method contribute the most. For the ablation study we use the best meta-embedding in the intrinsic and extrinsic evaluation tasks. We do this by skipping a different step of the method each time. For **-OOV** we do not apply the technique to obtain representations for the OOV words, we just average the available representations for a given word. With **-NORM** we do not perform the normalization steps to the source embeddings. For

FT	GV	W2V	J	UKB	P	AR	N	FJNP			
								CONC+	LLE	AAEME	Our Method
60.5	43.4	59.6	58.2	56.1	58.2	52.1	53.4	52.4	48.5	53.2	58.2

Table 6: Comparison of the source embeddings, our meta embedding method, baselines and previous work performance on GLUE benchmarks. GLUE score is reported.

-Vecmap the source embeddings are not mapped to a common vector space. The results reported in Table 7 show that the normalization and the mapping steps provide most of the performance. If we average embeddings that have not been normalized the difference in scale and the centroid of the vector space can cause some embeddings to take higher importance in the meta-embeddings. Averaging word embeddings that have not been mapped to the same vector space can cause vectors to cancel each other.

With respect OOV, the results are mixed. This step increases the performance in the categorization and word similarity tasks but it hurts the performance on the analogy and extrinsic tasks. This is caused by two factors. First, since all the embeddings have been normalized and mapped to the same vectors space, the average of the available representations is already a good approximation for OOV words. If the source embeddings would have a representation for the OOV words, it would be close to the ones already available.

Additionally, a larger vocabulary is not beneficial for every task. Consider the example in Table 2 where a much larger vocabulary obtains worse results in the Word Analogy task. We demonstrate this by counting the number of nearest neighbors to *love* with a cosine similarity greater than 0.85 in the meta-embeddings. Table 8 shows the most similar words when using and not using the OOV algorithm (27 and only 4 words respectively). Generating a meta-embedding containing the union of the vocabularies of all the source embeddings may be useful for some tasks, such as word similarity. However, for tasks such as word analogy, reducing the final vocabulary to the set of most common words is the best approach.

FJNP	AVG	C	WS	A	GLUE
Our method	**70.6**	73.5	**78.4**	59.9	58.2
-OOV	**70.6**	72.5	78.1	**61.2**	**59.5**
-NORM	67.5	**73.9**	77.0	51.7	55.3
-Vecmap	66.7	72.7	75.0	52.3	58.0

Table 7: Ablation studies on our standardization steps.

with OOV : **overlove**, **outlove**, **antilove**, **loveaholic**, have_no_regrets, sometimes_good, wonderful_feeling, strong_like, filial_love, lovedom, propose_to_woman, lov?d, family_love, Love, love_dearly, love_heart, Adore, buy_ring, LOVE, deep_affection, being_in_love, sovietophile, loveful, Loving, mislove, lovemonger, arachnophile
without OOV : **overlove**, **loveaholic**, **outlove**, **antilove**

Table 8: Nearest neighbors to the word *love* for the FJNP meta-embedding with a cosine similarity > 0.85 applying or not the OOV generation algorithm.

7 Conclusions

We have presented a meta-embedding generation method that improves over previous approaches. Moreover, our method does not rely on hyperparameter tuning and generates general-purpose meta-embeddings that can be used for any task. We also propose a comprehensive and unified evaluation framework for evaluating meta-embeddings. This framework allows to fairly and objectively compare different meta-embedding generation approaches using the same settings and methodology.

Using this framework we demonstrate that combining embeddings that encode the most complementary knowledge produces better meta-embeddings. In fact, the meta-embeddings that encode in the same vector space the knowledge from large text corpora, WordNet, PPDB and ConceptNet achieve the best published results in the intrinsic evaluation benchmarks. Interestingly, and contrary to what previous research suggested, we empirically demonstrate that when evaluating in a large set of extrinsic tasks, meta-embeddings are not helpful for improving the results of the source embeddings. We plan to investigate the performance of our approach in a cross-lingual setting for under-resourced languages. We suspect that the performance of under-resource language embeddings can be improved by combining them with embeddings from a rich-resource language.

Acknowledgments

We are grateful to the anonymous reviewers for their insightful comments. This work has been partially funded by the Spanish Ministry of Science, Innovation and Universities (DeepReading RTI2018-096846-B-C21, MCIU/AEI/FEDER, UE) and DeepText (KK-2020/00088), funded by the Basque Government. Iker García-Ferrero is supported by a doctoral grant from the Basque Government (PRE_2020_2_0208). Rodrigo Agerri is also funded by the RYC-2017–23647 fellowship. We also acknowledge the donation of a Titan V GPU by the NVIDIA Corporation.

References

Abdulrahman Almuhareb and Massimo Poesio. 2005. Concept learning and categorization from the web. In *proceedings of the annual meeting of the Cognitive Science society*, volume 27.

Kushal Arora, Aishik Chakraborty, and Jackie C. K. Cheung. 2020. Learning lexical subspaces in a distributional vector space. *Transactions of the Association for Computational Linguistics*, 8:311–329.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019. An effective approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 194–203, Florence, Italy. Association for Computational Linguistics.

Amir Bakarov. 2018. A survey of word embeddings evaluation methods. *CoRR*, abs/1801.09536.

Roy Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second PASCAL recognising textual entailment challenge. *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10, Edinburgh, UK. Association for Computational Linguistics.

William F Battig and William E Montague. 1969. Category norms of verbal items in 56 categories a replication and extension of the connecticut category norms. *Journal of experimental Psychology*, 80(3p2):1.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. *TAC*.

Danushka Bollegala and Cong Bao. 2018. Learning word meta-embeddings by autoencoding. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1650–1661, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Danushka Bollegala, Kohei Hayashi, and Ken-ichi Kawarabayashi. 2018. Think globally, embed locally - locally linear meta-embedding of words. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 3970–3976. ijcai.org.

Danushka Bollegala, Alsuhaibani Mohammed, Takanori Maehara, and Ken-ichi Kawarabayashi. 2016. Joint word representation learning using a corpus and a semantic lexicon. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2690–2696. AAAI Press.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*. AAAI Press.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of artificial intelligence research*, 49:1–47.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Wenfan Chen, Mengmeng Sheng, Jiafa Mao, and Weiguo Sheng. 2020. Investigating word meta-embeddings by disentangling common and individual information. *IEEE Access*, 8:11692–11699.

Joshua Coates and Danushka Bollegala. 2018. Frustratingly easy meta-embedding – computing meta-embeddings by averaging source word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 194–198, New Orleans, Louisiana. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 160–167. ACM.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer.

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.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Yerai Doval, Jose Camacho-Collados, Luis Espinosa-Anke, and Steven Schockaert. 2018. Improving cross-lingual word embeddings by meeting in the middle. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 294–304, Brussels, Belgium. Association for Computational Linguistics.

Philipp Dufter, Nora Kassner, and Hinrich Schütze. 2021. Static embeddings as efficient knowledge bases? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2353–2363, Online. Association for Computational Linguistics.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado. Association for Computational Linguistics.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: the concept revisited. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*, pages 406–414. ACM.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia. Association for Computational Linguistics.

Iker García-Ferrero, Rodrigo Agerri, and German Rigau. 2020. A common semantic space for monolingual and cross-lingual meta-embeddings. *ArXiv preprint*, abs/2001.06381.

Sahar Ghannay, Benoit Favre, Yannick Estève, and Nathalie Camelin. 2016. Word embedding evaluation and combination. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 300–305, Portorož, Slovenia. European Language Resources Association (ELRA).

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

Josu Goikoetxea, Eneko Agirre, and Aitor Soroa. 2016. Single or multiple? combining word representations independently learned from text and wordnet. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2608–2614. AAAI Press.

Josu Goikoetxea, Aitor Soroa, and Eneko Agirre. 2015. Random walks and neural network language models on knowledge bases. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1434–1439, Denver, Colorado. Association for Computational Linguistics.

Josu Goikoetxea, Aitor Soroa, and Eneko Agirre. 2018. Bilingual embeddings with random walks over multilingual wordnets. *Knowledge-Based Systems*, 150:218–230.

Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. Large-scale learning of word relatedness with constraints. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 1406–1414. ACM.

Jingyi He, Kc Tsiolis, Kian Kenyon-Dean, and Jackie Chi Kit Cheung. 2020. Learning efficient task-specific meta-embeddings with word prisms. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1229–1241, Barcelona, Spain (Online). International Committee on Computational Linguistics.

F Hill, Kyunghyun Cho, Sebastien Jean, C Devin, and Yoshua Bengio. 2014. Not all neural embeddings are born equal. In *NIPS 2014 Workshop on Learning Semantics*.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Stanislaw Jastrzebski, Damian Lesniak, and Wojciech Marian Czarnecki. 2017. How to evaluate word embeddings? on importance of data efficiency and simple supervised tasks. *CoRR*, abs/1702.02170.

Pratik Jawanpuria, Satya Dev N T V, Anoop Kunchukuttan, and Bamdev Mishra. 2020. Learning geometric word meta-embeddings. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 39–44, Online. Association for Computational Linguistics.

David Jurgens, Saif Mohammad, Peter Turney, and Keith Holyoak. 2012. SemEval-2012 task 2: Measuring degrees of relational similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 356–364, Montréal, Canada. Association for Computational Linguistics.

Douwe Kiela, Changhan Wang, and Kyunghyun Cho. 2018. Dynamic meta-embeddings for improved sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1466–1477, Brussels, Belgium. Association for Computational Linguistics.

Alexandros Komninos and Suresh Manandhar. 2016. Dependency based embeddings for sentence classification tasks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1490–1500, San Diego, California. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Hector J Levesque, Ernest Davis, and Leora Morgenstern. 2011. The Winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, volume 46, page 47.

Qichen Li, Xiaoke Jiang, Jun Xia, and Jian Li. 2020. Meta-embeddings based on self-attention. *ArXiv preprint*, abs/2003.01371.

Ken McRae, George S Cree, Mark S Seidenberg, and Chris McNorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior research methods*, 37(4):547–559.

Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Tomás Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

George A. Miller. 1992. WordNet: A lexical database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. 2017. Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association for Computational Linguistics*, 5:309–324.

James O'Neill and Danushka Bollegala. 2020. Meta-embedding as auxiliary task regularization. In *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de*

Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020), volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2124–2131. IOS Press.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Mohammad Taher Pilehvar, Dimitri Kartsaklis, Victor Prokhorov, and Nigel Collier. 2018. Card-660: Cambridge rare word dataset - a reliable benchmark for infrequent word representation models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1391–1401, Brussels, Belgium. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Vikas Raunak. 2017. Simple and effective dimensionality reduction for word embeddings. In *"Learning with Limited Labeled Data: Weak Supervision and Beyond" workshop at NIPS*.

Herbert Rubenstein and John B Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Alexandre Salle, Aline Villavicencio, and Marco Idiart. 2016. Matrix factorization using window sampling and negative sampling for improved word representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 419–424, Berlin, Germany. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. BERTRAM: Improved word embeddings have big impact on contextualized model performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3996–4007, Online. Association for Computational Linguistics.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, Lisbon, Portugal. Association for Computational Linguistics.

Yang Shao. 2017. HCTI at SemEval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 130–133, Vancouver, Canada. Association for Computational Linguistics.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 151–161, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

Robyn Speer and Joanna Lowry-Duda. 2017. ConceptNet at SemEval-2017 task 2: Extending word embeddings with multilingual relational knowledge. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 85–89, Vancouver, Canada. Association for Computational Linguistics.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments.

Transactions of the Association for Computational Linguistics, 7:625–641.

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

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Wenpeng Yin and Hinrich Schütze. 2016. Learning word meta-embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1351–1360, Berlin, Germany. Association for Computational Linguistics.

A Meta-embedding Generation Algorithm Illustrated

In this section we illustrate our meta-embedding generation algorithm using two sample embeddings with 3 dimension vectors and 1000 word vocabulary sizes (Figure 1). The vocabularies of the two embeddings have 791 common words, and each embedding has 209 unique words for which the other embeddings does not have a representation (OOV words). The resulting meta-embedding vocabulary will be the union of the vocabularies, 1197 words. Our approach to generate meta-embeddings consists of two main steps (i) pre-processing of the source embeddings and (ii) generation of the meta-embedding by averaging.

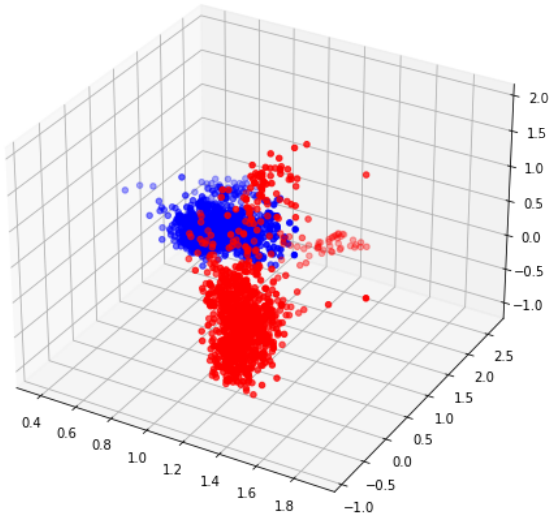

Figure 1: **Step 0** Source embeddings at the start of the embedding generation process

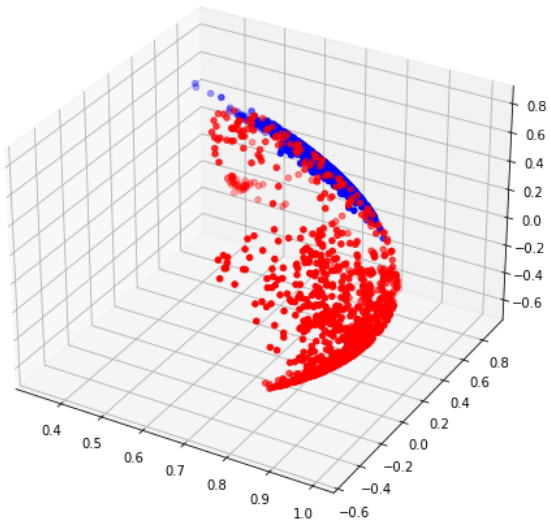

Figure 2: **Step 1** Length normalization of the source embeddings

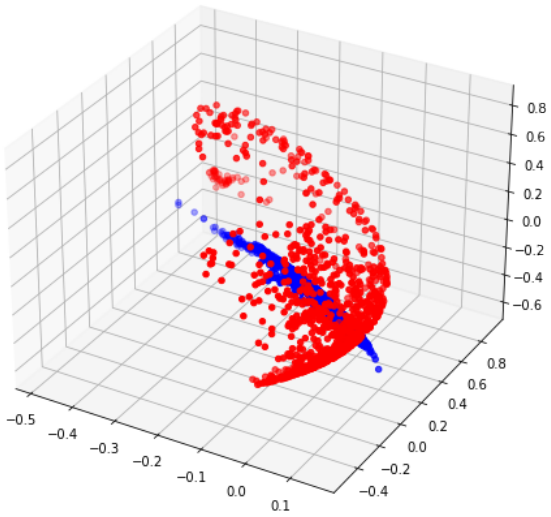

Figure 3: **Step 2** Mean centering of the source embeddings

A.1 Word embeddings pre-processing

Word embedding generated with different sources or techniques can result in very different vectors spaces and vocabularies. Before aligning the vector spaces an harmonization pre-processing step is needed. Thus, we translate, scale, rotate and match the vocabularies of the source embeddings.

1) Mean Centering and scaling: Following (Artetxe et al., 2018) we first length normalize the source embeddings (Figure 2). We mean center each dimension (Figure 3), and we length normalize them again (Figure 4). This translates all the source embeddings to the origin and scales them to have the same length.

2) Aligning the vector spaces: We align the vector spaces of the source embeddings using VecMap (Artetxe et al., 2016) (Figure 5). VecMap learns word embedding mapping using an orthogonal transformation. Orthogonality allows monolingual invariance during the mapping, preserving vector dot products between word vectors. Monolingual invariance ensures no information loss during the mapping step, which is desirable for our aim of generating meta-embeddings. In our experiments we align the source embedding by projecting them to the vector space of one particular source embeddings involved in the construction of

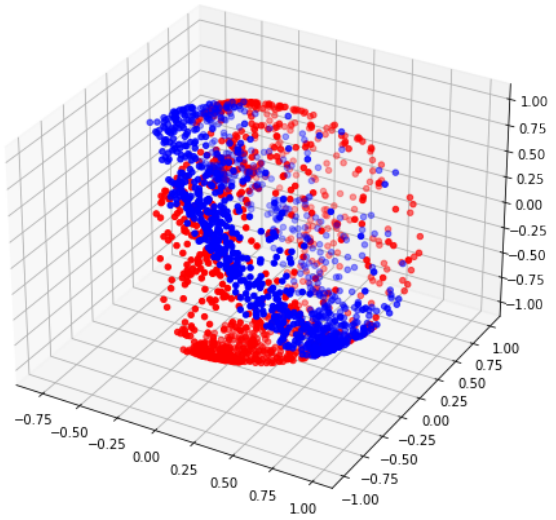

Figure 4: **Step 3** Second length normalization of the source embeddings

the meta-embeddings.

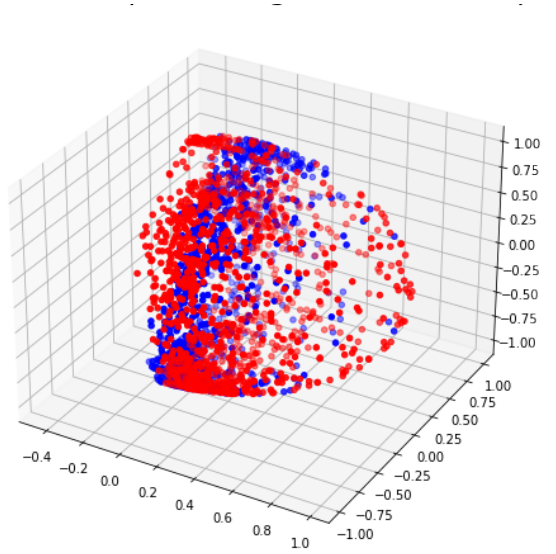

Figure 5: **Step 4** Alignment of the source embeddings using VecMap

3) OOV generation: Different word embeddings have different vocabularies. When combining two word embeddings we can distinguish two sets of words. Those for which we have a representation in both embeddings and those for which one of the embeddings has no representation. We call the latter "OOV words". We unify the vocabulary of the source embeddings by creating new approximate representations for the OOV words (Figure 6). The process is as follows. Given two source embeddings E1 and E2 where for a word W only E1 has a representation, we generate a new approximation for the OOV word in E2 by revising the most similar words from the common vocabulary of E1 and E2. First, using the cosine similarity as distance metric, we select the k (ranging from 2 to 50) nearest neighbours of the word W in E1 that also appear in the common vocabulary with E2. For each k, we calculate k candidate representations of the OOV word in E2 and E1 as a weighted average of the selected k nearest neighbours in their corresponding spaces. We use the cosine similarity from the nearest neighbors in E1 to W as weights. Finally, the selected representation of the OOV word in E2 is the one corresponding to the closest candidate to W in E1.

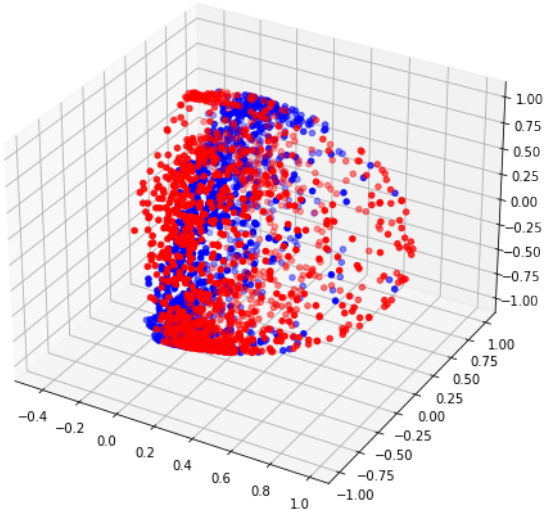

Figure 6: **Step 5** OOV generation algorithm

A.2 Meta-embedding generation

We combine the harmonized source embeddings by averaging them (Figure 7). We empirically demonstrate that thanks to the pre-processing steps, averaging source embeddings effectively combines multiple source embeddings resulting in representations as good as the ones generated by embedding concatenation without increasing its dimensionality.

B Computing infrastructure

We run all the experiments in a Linux system with an Intel Xeon CPU E5-2620 V4 CPU, 1024GB of RAM and an Nvidia Titan V GPU. To reproduce the generation of the FJNP meta-embedding with a reasonable run-time (less than 24 hours) we recommend using at least a quad-core CPU, 32GB of

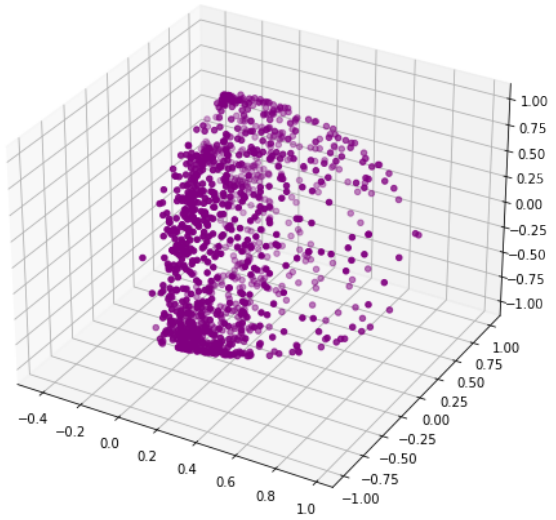

Figure 7: **Step 6** Meta-embedding generation by averaging

RAM and a 2GB GPU with CUDA support (GPU is optional but highly recommended). The intrinsic evaluation framework can be run in less than one hour in a system with enough primary memory to load a full embedding/meta-embedding (8GB). The extrinsic evaluation framework will run in less than 24 hours in a system with a reasonably modern CPU and enough primary memory to load the full embedding/meta-embedding and the bag-of-words model (8GB). The extrinsic evaluation can be speed-up with an 8GB GPU with CUDA and FP16 support.