Natural Language Engineering (2023), **29**, pp. 643–668
doi:10.1017/S1351324922000328

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

From unified phrase representation to bilingual phrase alignment in an unsupervised manner

Jingshu Liu[1,2,*], Emmanuel Morin[1,2], Sebastian Peña Saldarriaga[2] and Joseph Lark[2]

[1]LS2N – UMR CNRS 6004, Université de Nantes, Nantes, France and [2]Dictanova, 6 rue René Viviani, 44200 Nantes, France
*Corresponding author. E-mail: liujish931@hotmail.com

(Received 2 March 2020; revised 20 June 2022; accepted 27 June 2022; first published online 1 August 2022)

Abstract

Significant advances have been achieved in bilingual word-level alignment, yet the challenge remains for phrase-level alignment. Moreover, the need for parallel data is a critical drawback for the alignment task. This work proposes a system that alleviates these two problems: a unified phrase representation model using cross-lingual word embeddings as input and an unsupervised training algorithm inspired by recent works on neural machine translation. The system consists of a sequence-to-sequence architecture where a short sequence encoder constructs cross-lingual representations of phrases of any length, then an LSTM network decodes them w.r.t their contexts. After training with comparable corpora and existing key phrase extraction, our encoder provides cross-lingual phrase representations that can be compared without further transformation. Experiments on five data sets show that our method obtains state-of-the-art results on the bilingual phrase alignment task and improves the results of different length phrase alignment by a mean of **8.8** points in MAP.

Keywords: Multilinguality; Phrase representation; Bilingual alignment; Word semantics

1. Introduction

Bilingual phrase alignment from comparable corpora is the task of making explicit translation equivalent relations that exist between the phrases of two texts without a source text–target text relationship. Unsupervised bilingual phrase alignment is difficult. In this work, a phrase refers to single words and multiword expressions of any type, such as nominal or verbal phrases. Hence, the first challenge consists in learning a unified phrase representation, so that phrases can be compared independently of their length. The second challenge is the alignment, which cannot be solved directly without supervised cross-lingual information.

In this work we tackle these two challenges, first, we propose a method for learning a length-independent phrase representation, then we integrate this method into an end-to-end training architecture to learn bilingual representations in an unsupervised manner. Consequently, bilingual phrase alignment becomes a vector comparison task using the bilingual representations previously learned.

1.1 Unified phrase representation

Learning unified phrase representation can be seen as a short sequence language modeling task with one special property: the modeled representation should be one single unit (e.g., one vector) for inputs of variable length. As with a long sequence (e.g., a sentence) modeling task, both the

© The Author(s), 2022. Published by Cambridge University Press.

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

compositionality and the hierarchical syntactic relation of the composing words should be taken into consideration. For instance, although most phrases are freely combined like in "wind turbine" and "life quality", the meaning of some idiomatic or semi-idiomatic phrases can diverge from their constituent words as in "couch potato". Besides, even among more compositional phrases, the inner syntactical structure determines how constituent words are connected, hence influencing the overall semantics. For example, in the compound noun "sneaker shoe", the "sneaker" constituent dominates the semantics of the phrase when associated with "shoe".

Naively, we can pretrain phrase embeddings if we consider them as a single token, but it ignores compositionality and inner component relations of the phrase. Furthermore, learning phrase embeddings as individual vocabulary entries is extremely memory intensive and will lead to a data sparsity problem. Finally, phrases not seen during training cannot be handled by this approach. Artetxe, Labaka, and Agirre (2018b) proposed a generalized skip-gram that learns n-gram embeddings on-the-fly while keeping the desirable property of unigram invariance to handle compositional phrases but it still suffers from the sparsity and memory problem. Regarding the compositional method, two major approaches have been exploited in previous works. They both use word-level vectors for composing phrase representation. The first one consists in simple linear functions like addition, element-wise multiplication, or concatenation (Mitchell and Lapata 2009; Mikolov *et al.* 2013b; Garten *et al.* 2015; Goikoetxea, Agirre, and Soroa 2016; Hazem and Daille 2018; Liu, Morin, and Peña Saldarriaga 2018). The first two vector combination methods are simple and proved to be very effective in many NLP tasks; however, they ignore the syntactical structure of the phrase. In other words, these methods do not distinguish the word order. For example, "service department" and "department service" will have the same representation while they do not convey equal semantics. The concatenation method does register word order but variable length phrases are no longer semantically comparable even if we pad them. In addition, they all ignore the inner structure of the phrase. The second family of approaches includes more complex information, as they usually involve neural networks trained with extra information such as the phrase textual context (the words before and after the phrase) or a syntax tree structure (a part-of-speech parsing tree). Several works (Socher, Manning, and Ng 2010; Socher *et al.* 2013b; Irsoy and Cardie 2014; Paulus, Socher, and Manning 2014; Le and Zuidema 2015) have had promising results by using recursive neural networks (Goller and Küchler 1996) to capture syntactical information. However, the recursive neural network requires a tree structure for each training sample which may not always be available. To address this limitation, we propose a new tree-free recursive neural network to encode phrases of variable length into a single vector while preserving the compositionality and the syntactic information within the phrase.

More recently, contextualized word representations (Peters *et al.* 2018; Devlin *et al.* 2018; Radford *et al.* 2019) have achieved appealing improvements on a range of NLP tasks, but the models are mainly evaluated in classification-like or span prediction tasks, whether on sequence- or token-level (Rajpurkar *et al.* 2016; Wang *et al.* 2018; Williams, Nangia, and Bowman 2018; Zellers et al. 2018). While our final task, bilingual phrase alignment, is a vector comparison task, we would like to evaluate these approaches on similar comparative tasks such as the phrase synonymy or similarity by extracting the vectors calculated by the previous layers before the final classification layer. Note that these models generate a sequence of vectors so we still have to apply some extra procedures to retrieve one single fixed-length vector to represent the whole sequence in our scenario. The two major strategies are also eligible: we could simply use the mean vector over the sequence or choose one specific vector to represent all the sequence, otherwise we could stack other neural networks which can generate a single fixed-length vector given a sequence input.

Our tree-free recursive neural network can be trained in a typical encoder–decoder architecture exploited in many neural machine translation frameworks or a Siamese-like system (Bromley *et al.* 1993). The advantage of these end-to-end systems is that they can be easily scaled up or incorporated in other networks without extra training information.

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

We evaluate our system on four data sets: two open domain data sets from Semeval 2013 and 2017 and two specialized domain data sets. The first corpus, from a European public project, covers the renewable energy domain in English and French, while the second will be released with this paper and covers a cancer subtopic in the medical domain. The results obtained improve state-of-the-art approaches on the similarity and synonymy tasks. Furthermore, several ablation tests are conducted to evaluate the impact of our phrase encoder, its training objective, and contextualized embeddings used as input.

1.2 Unsupervised bilingual phrase alignment

Beginning with the seminal works of Fung (1995) and Rapp (1999) based on word co-occurrences for BWA (bilingual word alignment), significant improvements have been recently achieved by neural network-based approaches (Mikolov, Le, and Sutskever 2013a; Faruqui and Dyer 2014; Xing *et al*. 2015; Artetxe, Labaka, and Agirre 2018a; Peng, Lin, and Stevenson 2021), but most work on the subject focuses on single terms. The alignment of multiword expressions (MWE) from comparable corpora is discussed less (Robitaille *et al*. 2006; Morin and Daille 2012). Our work is in line with Liu *et al*. (2018), where the objective is to rank all the candidates in a given list containing phrases of variable length based on a source phrase. Moreover, unlike Liu *et al*. (2018), our work can align phrases in an unsupervised manner without explicit cross-lingual information.

We adapt our tree-free recursive neural network as a phrase encoder for bilingual phrase alignment tasks because it can generate one single-fixed length vector for phrases of variable length while conserving the syntactical relation between words.

Concerning the model training, since the meaning of domain-specific phrases is highly context related, the commonly used sequence-to-sequence systems better fit our needs. After phrase encoding, we can decode its representation to predict its context, thus establishing a relation between the phrase and its context. Unlike common neural machine translation sequence-to-sequence systems, our model encodes a phrase and decodes it with regard to its syntactic context via our tree-free recursive neural network. In order to be able to align phrases in different languages, we make the encoder cross-lingual which means that the input vectors in different languages share the same vector space (Artetxe *et al*. 2018a; Liu *et al*. 2018). We also incorporate a back-translation mechanism (Sennrich, Haddow, and Birch 2016) of single words during training by using pretrained bilingual word embeddings (BWE). Moreover, our model relies exclusively on monolingual data, and is trained in an unsupervised manner. After completion of the training phase, we obtain a shared cross-lingual phrase encoder that can generate a unified representation of phrases of any length.

As for the data sets, we use the same specialized corpora as in our monolingual evaluation: one covers the renewable energy domain and the other covers the cancer subtopic of the medical domain. We manually create 3 gold standards for the first domain with 3 different language pairs: English-Spanish, English-French and English-Chinese, and 1 English-Spanish gold standard for the medical domain corpus. Our experiments on these data sets show that our method significantly outperforms existing unsupervised methods for the different length phrase alignment by a mean of 8.8 MAP points.

2. Background

2.1 Sequence representation modeling

The simple additive approach for encoding a sequence of word vectors into one single vector is always considered as an effective baseline (Mikolov *et al*. 2013b; Del, Tättar, and Fishel 2018; Liu *et al*. 2018; Laville *et al*. 2020; Huang *et al*. 2020). Another possible improvement is to use a recursive neural network (RNN) (Goller and Küchler 1996). It is a generalized version of the recurrent

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

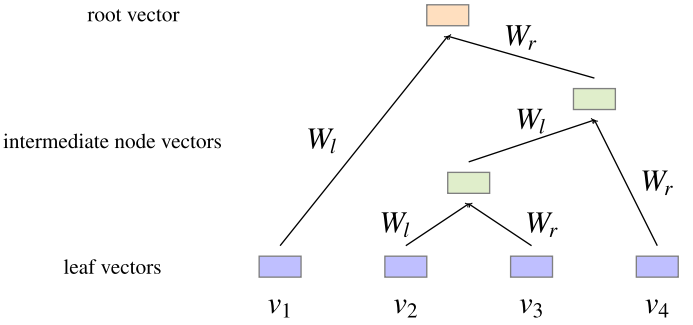

Figure 1. Diagram of a recursive neural network.

neural network (Elman 1990) which always applies a left binary tree, where the first two leaves are combined to form a node, then the node is combined to the next leaf to form the next level node, etc. The recursive neural network encodes a sequence of word vectors along a tree structure, for example a parse tree, by recursively applying the weight matrices to each node association. This architecture has been successfully exploited in a variety of tasks, Socher *et al.* (2013a) use an *untied weight* RNN for constituent parsing where they use different weight matrices depending on the constituent syntactic category, Le and Zuidema (2014) collect the context information by adding an outer representation for each node. Their system is served in a dependency parsing task. Moreover, various works (Socher *et al.* 2013b; Irsoy and Cardie 2014; Paulus *et al.* 2014) apply the RNN to generate sentence-level representations for the sentiment analysis task using labeled data.

Figure 1 shows an example of a sequence of length four. Suppose we have a parse tree, each input is a word vector $v_i \in \mathbb{R}^d$. The network applies a linear function with a weight matrix $W_l \in \mathbb{R}^{d*d}$ for each left node child and a weight matrix $W_r \in \mathbb{R}^{d*d}$ for each right node child in the given tree. So, for each non-leaf node η, the corresponding vector x_η is calculated as follows:

$$x_\eta = W_l v_{l(\eta)} + W_r v_{r(\eta)} + b \tag{1}$$

where $v_{l(\eta)}$ and $v_{r(\eta)}$ mean, respectively, the left and the right child vector of the node η.

The disadvantage of RNN in our scenario is the need of a tree structure because, as stated above, not only it is not always available in all languages but it is also not possible to retrieve the context sentence for the parsing if we meet a new freely combined phrase that has never occurred in the corpus. The recurrent neural network or the LSTM does not need a tree structure but applies a universal left binary tree to all sequences, The convolutional neural network, however, with a kernel size of 2 can be considered as a specialized RNN where it adopts element-wise multiplication rather than matrix multiplication with only one layer by a pooling operation. The more advanced and purely self-attention-based model, Multi-Head Attention cell (Vaswani *et al.* 2017) has shown great potential in sequence modeling, but it has many more parameters compared to the previously mentioned models. It is obvious that if a model has more parameters, it is more powerful. As a consequence, it is not directly comparable to other models that have significantly fewer learnable parameters.

Figure 2 shows how the recurrent and a 2 kernel sized convolutional neural network can model a sequence with 4 tokens. $W_{xh} \in \mathbb{R}^{h*d}$ and $W_{hh} \in \mathbb{R}^{h*h}$ are the parameters in a typical recurrent neural network where h is the hidden dimension, and for the convolutional network with a kernel size of 2, we can consider the convolution operation as two element-wise multiplications (dashed line in Figure 2) with a left multiplier $v_l \in \mathbb{R}^d$ and a right multiplier $v_r \in \mathbb{R}^d$. Stacking v_l and v_r forms the actual convolution kernel. The final vector is obtained by a pooling operation such as max or average. Note that the addition-based approach (Liu *et al.* 2018) can be viewed as a specialized version of CNN where the values in v_l and v_r equal to one and pooling is done by averaging the vectors.

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

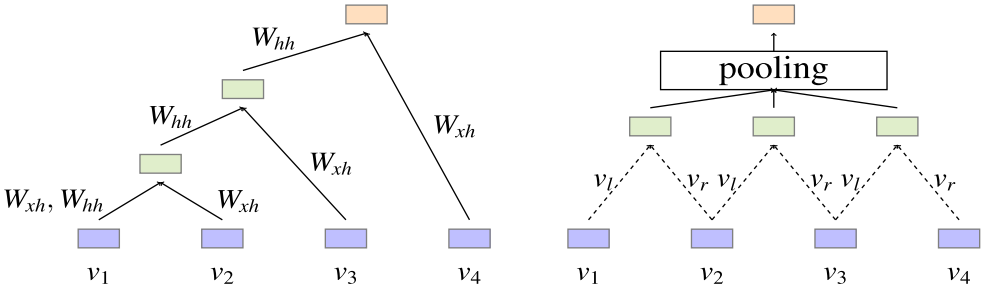

Figure 2. Diagram of a recurrent neural network (left) and 2 kernel sized convolutional neural network (right). For the purpose of clarity, we omit the output layer in the recurrent neural network.

Recently, published language models like BERT (Devlin *et al.* 2018) or ELMo (Peters *et al.* 2018) can also encode a sequence of word vectors of a phrase into one single vector if we use the output vector at one particular step, for example, the last token in ELMo or the first special [CLS] token in BERT. The additive approach can also be applied to all the output vectors. Like the static word embedding models, these models are pretrained on large general corpora. By default, they all encode a sequence of word vectors into a new sequence of word vectors that are also known as contextualized word embeddings. Consequently, the tasks similar to sequence labeling (e.g., sentence tagging) and span prediction (e.g., question answering) naturally fit these models. The sequence-level classification task can be achieved by representing the sequence with the output vector of a special token position like the first or last token of a sequence. To the best of our knowledge, these models have been applied mostly on classification or span prediction tasks.

2.2 Feature-based and fine-tuning-based language representations

The feature-based strategy has existed since the 1990s. The traditional co-occurrence count-based method (Church and Hanks 1990; Dagan, Pereira, and Lee 1994; Niwa and Nitta 1994; Bullinaria and Levy 2007; Turney and Pantel 2010) represents a word by a sparse co-occurrence vector and often applies the *pointwise mutual information* to associate the word and its context. Neural network-based methods (Mikolov *et al.* 2013b; Pennington, Socher, and Manning 2014) represent a word by a dense embedding vector, which has led to significant improvements in major NLP tasks. These word-level static vectors can be incorporated into other systems as the basic input units to generate higher level representations.

Contextualized models (Peters *et al.* 2018; Devlin *et al.* 2018; Radford *et al.* 2019) are sequence-level representations with word-level granularity. In fact, they all exploit word-level representations as the basic input and output units. Once pretrained, we can use these models in a specific task by stacking supplementary layers onto them. The difference between the feature based and the fine-tuning based approach lies in whether we freeze the parameters of these pretrained models or not when we incorporate them into a task-specific training framework. The feature-based approach extracts the output of the pretrained model and uses this output as static features of the input by omitting the gradients of the parameters, while the fine-tuning based approach updates its parameters during the back-propagation of the training. The advantage of the fine-tuning-based approach is that the whole system can be readjusted to the task-specific training corpus, but it takes up much more time and space consuming compared to the feature-based approach. Moreover, according to Devlin *et al.* (2018), similar performance can be obtained (with –0.3 points in F1 CoNLL-2003 NER) using the same BERT model in the feature and the fine-tuning based settings. This is particularly interesting because fine tuning a large model with millions of parameters can be exceedingly long while updating only a few layers is much more efficient.

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

2.3 Cross-lingual word embeddings

In order to map phrases of different languages into one common space with compositional models, word-level mapping is an essential prerequisite. Following the success of word embeddings (Mikolov *et al*. 2013b) trained on monolingual data, a large proportion of research concentrated on at mapping word embeddings into a common space for multiple languages. Cross-lingual word embeddings were pioneered by Mikolov *et al*. (2013a) by using a linear transformation matrix. A large number of works tried since then to improve the linear transformation method (Lazaridou, Dinu, and Baroni 2015; Artetxe, Labaka, and Agirre 2016; Liu *et al*. 2018). Artetxe *et al*. (2018a) compiled a substantial amount of similar works (Mikolov *et al*. 2013a; Faruqui and Dyer 2014; Xing *et al*. 2015; Shigeto *et al*. 2015; Zhang *et al*. 2016; Artetxe *et al*. 2016; Smith *et al*. 2017) into a multistep bilingual word embedding framework. More recently, Lample and Conneau (2019) proposed pretrained cross-lingual transformer-based language models using *masked language modeling* like Devlin *et al*. (2018) and a *translation language modeling* training objective with parallel data to further improve the quality of pretrained cross-lingual embeddings for languages that share the same alphabet.

2.4 Training objectives in language modeling

Predicting the next word or sentence is the most common training objective in a wide range of previous works with an architecture similar to encoder–decoder (Bahdanau, Cho, and Bengio 2014; Sutskever *et al*. 2014; Cho *et al*. 2014; Luong, Pham, and Manning 2015; Gehring *et al*. 2017; Vaswani *et al*. 2017; Peters *et al*. 2018; Devlin *et al*. 2018; Radford *et al*. 2019). In addition, the BERT model jointly has another training objective of predicting only the randomly masked tokens. The corresponding ablation studies have proven this to be beneficial. This objective can be considered as a special version of the denoizing objective (Vincent *et al*. 2008), which reconstructs the original sentence from a randomly noised version.

2.5 Encoder–decoders in neural machine translation with low-resource

To train our network, we use the widely exploited encoder–decoder model in neural machine translation (NMT). Although there are many different models, they all implement an encoder–decoder architecture optionally combined with an attention mechanism (Bahdanau *et al*. 2014; Luong *et al*. 2015) to tackle long sequences. This type of model has become the main trend in the recent years producing the current state-of-the-art results. It takes advantage of longer context information, and continuous representations can be easily trained in an end-to-end system.

Cho *et al*. (2014) proposed a model to learn representations of variable-length sequences, however, their approach requires parallel phrase pairs for training. Therefore, we looked at NMT models making use of monolingual corpora to enhance translation in low-resource scenarios. When no parallel data exist between source and target languages, several works proposed the use of a pivot language (Firat *et al*. 2016; Saha *et al*. 2016; Chen *et al*. 2017) acting as a bridge between source and target. Following the same idea, Johnson *et al*. (2017) proposed a multilingual NMT model that creates an implicit bridge between language pairs for which no parallel data are used for training. Whether explicit or implicitly, all these works still require the use of parallel corpora between the pivot language and other languages.

More interestingly for our work, some research has recently conducted on training NMT models with monolingual corpora only (Lample *et al*. 2018; Artetxe *et al*. 2018c; Yang *et al*. 2018). They all use pretrained cross-lingual word embeddings as input. Then a shared encoder is involved to encode different noised sequences in the source and the target languages. The decoder decodes the encoded vector to reconstruct its original sequence. This strategy is called *denoizing* (Vincent *et al*. 2008) with the objective to minimize the following cross-entropy loss:

$$\mathscr{L}_{denoising}(\theta_{enc}, \theta_{dec}) = -\mathbb{E}_{x \in D_l} H(x, dec_{\rightarrow l}(enc(\mathscr{N}(x)))) \tag{2}$$

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

where θ_{enc} and θ_{dec}, respectively, mean the parameters in the encoder and the decoder, $x \in D_l$ is a sampled sequence from the monolingual data, and $dec_{\rightarrow l}(enc(\mathcal{N}(x)))$ represents a reconstructed sequence from the noised version of the original sequence x.

In addition, the back-translation mechanism (Sennrich *et al.* 2016; Zhang and Zong 2016) has been dominantly exploited in unsupervised neural machine translation (Lample *et al.* 2018; Artetxe *et al.* 2018c) to build the link between the two languages by alternatively applying the source-to-target model to source sentences in order to generate inputs for training the target-to-source model (and vice versa):

$$\mathcal{L}_{backtranslation}(\theta_{enc}, \theta_{dec}) = -\mathbb{E}_{x \in D_{l1}} H(x, dec_{\rightarrow l1}(enc(y))),$$
$$y = transl(x) = dec_{\rightarrow l2}(enc(x)) \tag{3}$$

where D_{l1} and D_{l2} are the two language corpora, $dec_{\rightarrow l1}$ means that the decoder will decode the sequence in $l1$ language (or $l2$ resp.). Suppose y is the translation of $x \in D_{l1}$, then $dec_{\rightarrow l1}(enc(y))$ represents the reconstructed source sentence from the synthetic translation. The goal is to generate pseudo parallel sentence pairs to train the models with a reconstruction loss.

Also pertaining to our work, Yang *et al.* (2018) introduce a semi-shared encoder to retain specific properties of each language and directional self-attention to model word order. More recently, Wu, Wang, and Wang (2019) propose an alternative approach that extracts and edits candidate translation sentences with comparative loss.

To sum up, most of the previous works use either compositional approaches, for instance, the average of all word vectors of a sentence, or a representative vector of a neural network, for instance, the special CLS token in BERT to represent a sequence. While for the bilingual phrase alignment, the previous studies exploit essentially the cross-lingual word embeddings. This work is an extension of Liu *et al.* (2020) for the bilingual phrase alignment part which also uses cross-lingual word embeddings as input. Moreover, we provide full study on the phrase representation learning.

3. Unified phrase representation learning

3.1 Tree-free recursive neural network

In order to encode phrases of variable length without tree structures in a fixed-length vector, we propose a new network a *tree-free recursive neural network* (TF-RNN). We consider it as a variant of the original recursive neural network because the basic idea is always to associate each token following a bottom-up structure. This structure is required as input of the original recursive neural network, while in the TF-RNN we eliminate this requirement by recursively splitting each node into a left and a right semantic part, then associating the left part with its right-hand neighbor and the right part with its left-hand neighbor. This is motivated by our hypothesis that the semantics of a pair of words could be retrieved by combining their meaning with some position-specific weights, and consequently the semantics of a sequence of words could be retrieved by recursively combining the semantics of each word pair. In fact, by doing this, we create a pseudo binary tree structure where we associate each adjacent node pair without parsing it twice. This kind of structure can be seen as an approximation of a generalized sentence syntax as each language unit is directly associated with its adjacent neighbors and hierarchically associated with other units eventually yielding the overall semantics of all the units.

Let $\left[v_1^0, v_2^0, v_3^0, ..., v_n^0\right]$ with $v_i^0 \in \mathbb{R}^d$ be the input word vector sequence with n words, the TF-RNN outputs a single fixed-length vector $v_o \in \mathbb{R}^p$ by following steps:

$$v_{i,l}^j = \tanh(W_l v_i^j + b_l)$$
$$v_{i-1,r}^j = \tanh(W_r v_{i-1}^j + b_r)$$

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

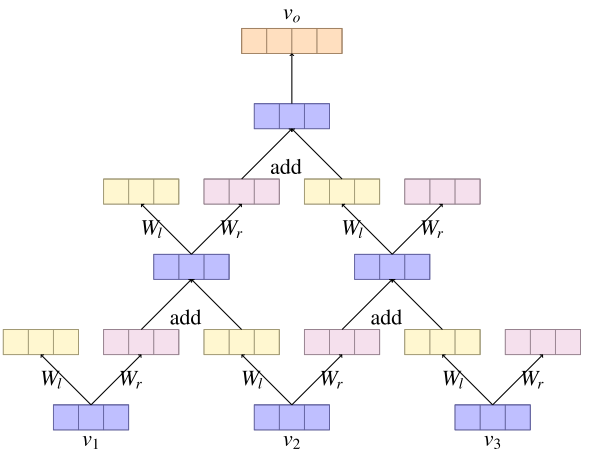

Figure 3. Diagram of the tree-free recursive neural network (TF-RNN).

$$
\begin{aligned}
v_i^{j+1} &= \tanh(v_{i,l}^{j} + v_{i-1,r}^{j}) \\
&\ldots \\
v_o &= \tanh(U v_0^n + b)
\end{aligned} \tag{4}
$$

where j indicates the pseudo-tree structure layer level, a phrase with n word components will have n levels in such a structure. $W_l \in \mathbb{R}^{d*d}$ and $W_r \in \mathbb{R}^{d*d}$, respectively, represent the left and right weight matrices for the extraction of the word semantics; $b_l \in \mathbb{R}^d$ and $b_r \in \mathbb{R}^d$ are the corresponding bias vectors. A node vector on level $j+1$, v_i^{j+1} is calculated in terms of a pair of adjacent node vectors from the previous level j. Once we reach the final level n, the final output vector v_o can be calculated by a linear layer on top with $U \in \mathbb{R}^{p*d}$ and $b \in \mathbb{R}^p$ as its parameters. A nonlinear activation function is applied after each operation. An example of a sequence of length three is illustrated in Figure 3.

3.2 Complexity

We compare the complexity of different neural network layers that can encode sequences of variable length, the RctNN and self-attention encode the input sequence to another sequence of equal length while our proposal TF-RNN and CNN with padding and pooling (right part of Figure 2) encode to one fixed-length vector.

We compare mainly two criteria for one layer of each architecture, the first is the computational complexity that represents how many weight parameters are involved in the linear transformation; the second is the maximum dependency length which is the length of the paths that forward and backward signals have to traverse in the network. This is critical for learning long-range dependencies in many sequence transduction tasks. We show the complexity comparison in Table 1.

For the computational complexity, the RctNN has n times linear transformations with matrices in $\mathbb{R}^{d*d}$, while the CNN is more expensive than RctNN by a factor of the kernel width k which can be seen as having k times the weight matrices in a RctNN. The self-attention is faster for most cases since usually $n < d$. Our proposal seems to be the most expensive since the complexity is quadratic in terms of input length and model dimension; however, it should be noted that our objective is to encode phrases which are most of the time n-gram with $n \in [1, 5]$. For unigrams

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

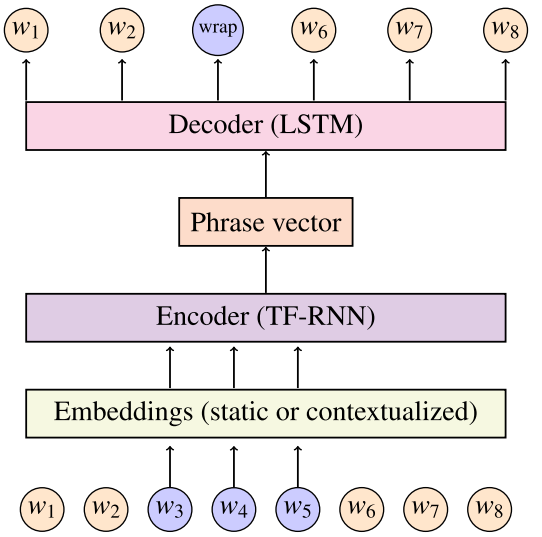

Figure 4. Proposed training system in monolingual context, w_i is the ith word in a sentence, *wrap* is the token represented by the randomly generated vector for filling the phrase blank when generating the context. In the example, [w_3, w_4, w_5] is the phrase sequence.

and bigrams, our encoder is less complex than RctNN and CNN. Also, our encoder is a one-layer architecture compared to the self-attention which has a "depth" of 8 heads in the Transformer-base architecture.

As the self-attention is a dynamic fully-connected layer, it traces each input position with one linear transformation. As for our proposal, we can see that it is linearly related to the input length, yet again since our inputs are mostly short sequences, this is not considered as problematic in our scenario.

3.3 Encoder–decoder training with wrapped context prediction

We use a fairly standard encoder–decoder architecture to train the phrase encoder. Our TF-RNN is used as encoder, thus phrases of variable length can be represented by a fixed-length vector without the need of a tree structure. The decoder is a two-layer LSTM. Furthermore, instead of predicting the next word or phrase like in many other similar systems, we let the generator generate the context of the phrase. However, one disadvantage of predicting only the context is that the syntax of the output sequence is misguided by the missing phrase. Since most of the phrases are either nominal or verbal, we decide to use a single random vector to wrap all the tokens of a phrase to help the generator reconstruct a syntactically complete context during the system training.

Apart from the static word embeddings, we also incorporate the recent works of contextualized embeddings as the input for our TF-RNN phrase encoder. As mentioned in Section 2.2, feature-based usage of these models degrades the results only slightly and the results compared to the fine-tuning based usage which takes up far more time and space. In our framework, we apply the feature-based approach to accelerate our experiments with relatively low resources. More concretely, this means that we freeze all the parameters in the contextualized embedding models when back-propagating the whole network. After the training, the encoder serves as a phrase vector generator which takes either static or contextualized embedding vectors. Figure 4 shows an instance of the framework.

Note that in the case of the static embeddings, the embedding layer is actually a look-up table while for the contextualized ones, it is a forward pass of the pretrained model. More specifically,

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

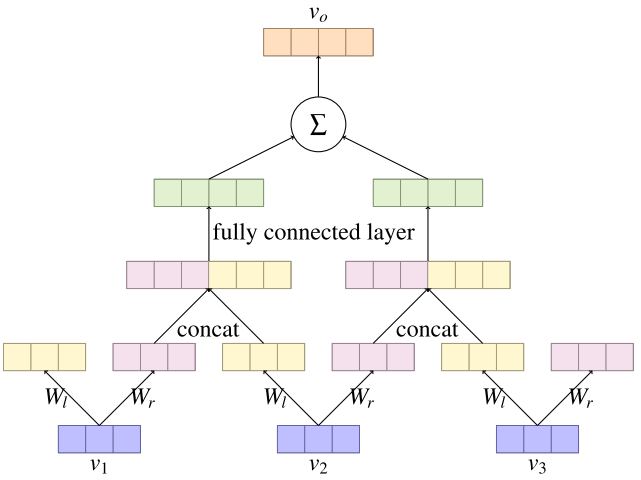

Figure 5. Illustration of the tree-free phrase encoder.

to get the word vector from the static embeddings, one can directly use one row or column of the hidden layer matrix with the word index. For the contextualized embeddings, it passes the entire sentence (or several words between the word) to the neural network and run a forward pass to get the context-dependant word vector.

4. Toward unsupervised bilingual phrase alignment

4.1 Tree-free phrase encoder in cross-lingual context

We would have used the same encoder explained in Section 3.3, however, in our preliminary experiments, it did not perform well as the synthetic translations are sometimes of low quality and the accumulated translation errors affect with the recursivity more radically (Wu *et al*. 2019). The same phenomenon occurs also in other similar networks such as the recurrent or LSTM network. Meanwhile, since the additive approach (Liu *et al*. 2018) manages to maintain a decent performance, we decided to adapt the tree-free recursive neural network to the cross-lingual context by levelling the network. Consequently, the network has more additive features while being able to distinguish the word order and distribute different weights. More concretely, there are three layers in the adapted version, in the first we always split the semantics of each word into two parts by a linear transformation: the right side and the left side. Then we associate these nodes by concatenation, the left side is supposed to be associated with the right side of the previous token and vice versa. The second layer is composed of a fully connected layer that maps the input vectors to output vectors in a specified dimension. Finally, the third layer consists in the addition of all intermediate level nodes and outputting a single fixed-length vector. The sum operation is motivated by the additive characteristics mentioned in Mikolov *et al*. (2013b) as the additive approach has showed interesting results in our preliminary experiments. Figure 5 shows the schema of the proposed network which is clearly a flat version of the TF-RNN presented in Section 3.1.

We use pretrained cross-lingual embeddings as the input vector sequence $[v_1, v_2, v_3, ..., v_n]$ with $v_i \in \mathbb{R}^d$, the output vector $v_o \in \mathbb{R}^p$ is calculated as follows:

$$\begin{aligned} v_{i,l} &= \tanh(W_l v_i + b_l) \\ v_{i-1,r} &= \tanh(W_r v_{i-1} + b_r) \\ v_{inter,i} &= \tanh(U[v_{i-1,r}; v_{i,l}] + b) \\ v_o &= \sum_{i=1}^{n} v_{inter,i} \end{aligned} \tag{5}$$

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

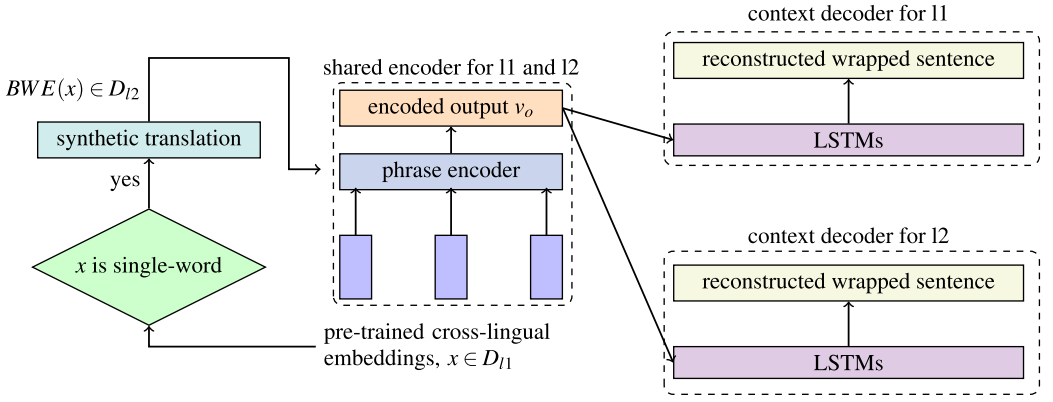

Figure 6. Overview of the cross-lingual alignment training architecture. For a phrase x in language D_{l1}, we first use the shared tree-free phrase encoder, then the system can be trained into two subnetworks: the first one is the encoder–decoder system given the original phrase x w.r.t D_{l1}, and if x is a single-word phrase, we apply a second encoder–decoder system given the translated phrase $BWE(x)$ also w.r.t D_{l1}. We alternatively iterate through all phrases in the two languages. The objective of the decoder is to reconstruct a *wrapped sentence* containing x.

where $W_l \in \mathbb{R}^{d*d}$ and $W_r \in \mathbb{R}^{d*d}$ denote the left and the right weight matrix, respectively, in the linear transformation of the semantic association, $b_l \in \mathbb{R}^d$ and $b_r \in \mathbb{R}^d$ are the corresponding bias vectors, *and* $U \in \mathbb{R}^{p*d}$ and $b \in \mathbb{R}^p$ are the parameters in the fully connected layer with d the input dimension and p the output vector dimension.

Consequently, our phrase encoder produces vector representations that are word order sensitive and that can distribute different weights for the different phrase components without using structured input.

4.2 Unsupervised training

The general encoder–decoder architecture of our method is shown in Figure 6. Since the input sequence is always a short sequence of under 7 tokens, usually a two or three word phrase, we did not use an attention mechanism which is intended to capture long-range dependencies. The network tries to predict the sentence containing the input phrase from its encoded vector. One can argue that our system is only unsupervised under the prerequisite of pretrained bilingual embeddings. This is true. However, since pretrained embeddings are largely available and can easily be obtained with general public parallel data, we consider that our system is unsupervised because we do not need specific parallel data.

As illustrated in Figure 6, in addition to our phrase encoder, we incorporate a pseudo back-translation mechanism for single words based on bilingual word embeddings (Artetxe *et al*. 2018a; Liu *et al*. 2018). The decoder consists of a single-layer LSTM and a fully connected layer on top of it. The goal of the decoder is to reconstruct the *wrapped sentence* which contains the current input phrase. We name this process *context prediction*. The intuition behind context prediction is based on the distributional hypothesis (Harris 1954), that is *words in similar contexts tend to have similar meanings*. This idea is studied in Del *et al*. (2018): instead of an end-to-end system, they first learn all the phrase embeddings by Skip-gram considering them as a single word, and then learn the composition function by a regression model that predicts the pretrained phrase embeddings from its composing word embeddings. However, they limit the phrase length to 2, while we would like to propose a unified end-to-end framework which is able to learn the phrase composition of variable length and the mapping simultaneously. Overall, the system uses three key concepts:

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

Wrapped sentences. Like in NMT, we use special tokens to mark the start and the end of a sentence. Apart from the standard special tokens, we exploit the same training strategy as in our monolingual system: the wrapped sentence. In addition to what we have stated in Section 2.4, this allows the system to recognize the phrase when decoding and to strengthen links between languages.

Shared encoder. The system treats input phrases in different languages via the universal encoder detailed in Section 4.1. Works using a similar idea are He *et al.* (2016), Lee, Cho, and Hofmann (2017) and Artetxe *et al.* (2018c). As the input embeddings are already mapped to a common space, the representation generated by the shared encoder is also a cross-lingual vector. After the training, we use exclusively the shared encoder to generate cross-lingual phrase representations, which is essential for our final task: bilingual phrase alignment.

Pseudo back-translation. Since we do not have cross-lingual data, a direct link between a phrase in language $l1$ and one in language $l2$ is not feasible. However, synthetic translations of single words can be easily obtained using bilingual word embeddings. By using translated single-word phrases to train our model, we create stronger links between the two languages. This can be viewed as pseudo *back-translation* as we generate synthetic translations by BWE while in NMT systems the translation is generated by the corresponding decoder (Sennrich *et al.* 2016; Artetxe *et al.* 2018c).

Therefore, the system potentially has four objective loss functions when we alternatively iterate all phrases in the two languages $l1$ and $l2$:

$$\mathscr{L}_{cp\ l1\to l1}\left(\theta_{enc}, \theta_{dec_{\to l1}}\right) = -\mathbb{E}_{x\in D_{l1}} H(ws(x), dec_{\to l1}(enc(x))), \tag{6}$$

$$\mathscr{L}_{cp\ l2\to l1}\left(\theta_{enc}, \theta_{dec_{\to l1}}\right) = -\mathbb{E}_{x\in D_{l1}} H(ws(x), dec_{\to l1}(enc(\mathrm{BWE}(x)))), \tag{7}$$

$$\mathscr{L}_{cp\ l2\to l2}\left(\theta_{enc}, \theta_{dec_{\to l2}}\right) = -\mathbb{E}_{x\in D_{l2}} H(ws(x), dec_{\to l2}(enc(x))), \tag{8}$$

$$\mathscr{L}_{cp\ l1\to l2}\left(\theta_{enc}, \theta_{dec_{\to l2}}\right) = -\mathbb{E}_{x\in D_{l2}} H(ws(x), dec_{\to l2}(enc(\mathrm{BWE}(x)))) \tag{9}$$

where $\mathscr{L}_{cp\ lp\to lq}$ means the *context prediction* loss from an encoded phrase in language lp to the context of language lq, $dec_{\to l}(enc(x))$ is the reconstructed version of the wrapped sentence, $ws(x)$ denotes the real wrapped sentence containing the phrase x and $\mathrm{BWE}(x)$ is the translated single-word phrase for x using bilingual word embedding.

5. Experiment settings

5.1 Phrase synonymy and similarity

Data and resources. For the phrase synonymy task, we use two specialized domain corpora: *Wind Energy* (WE)[a] and a new *Breast Cancer* (BC) corpus. The WE corpus comes with 6 languages. In this work, we only evaluate on the English and the French corpora which have, respectively, 13,338 and 33,887 sentences. The BC corpus is in English and contains 26,716 sentences. The aim of the phrase synonymy task is to find a phrase synonym in a given corpus. Usually a large list of candidates is first extracted from the corpus so we can select candidates which are the most likely phrase synonyms. In order to build the candidate list, we use the *IXA pipes* (Agerri *et al.* 2014) library[b] to preprocess the corpora with the built-in preprocessing tools following this order: normalization, tokenization and pos-tagging. Then a list of phrases of a maximum of 7 words is extracted using the open source tool PKE.[c] Finally, 8923 and 6412 phrases are extracted from

[a] https://www.ls2n.fr/ressources-langagieres.
[b] http://ixa2.si.ehu.es/ixa-pipes/.
[c] https://github.com/boudinfl/pke.

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

the English and French WE corpora and 8989 phrases from the BC corpus after filtering hapaxes (threshold 1). We use the same gold standard as Hazem and Dailles (2018) for the WE corpus. The gold standard for the BC corpus was built based on the MeSH 2018 thesaurus[d] and contains 108 phrases.

As for the phrase similarity task, two public open domain data sets are obtained from the task 2 of previous *Semeval* campaigns. One is from task 2 of *Semeval 2017* (Camacho-Collados *et al.* 2017)[e] and the other one from the task 5 of *Semeval 2013* (Korkontzelos *et al.* 2013).[f] The *Semeval 2017* data set has a gold standard of 95 pairs of phrases after filtering those with only single words, an evaluation script, a 64-dimensional static word embedding model and a wiki corpus with 46 million sentences which contains the context information of the phrases. The *Semeval 2013* data set only contains a gold standard of 7814 pairs of multiword phrases.

Input embeddings. Regarding the embedding model, we use *deeplearning4j*[g] to train domain-specific 100-dimensional word embeddings using the Skip-gram model, with 15 negative samples and a window size of 5. Since the specialized corpora are fairly small, we concatenate these embeddings with the 300-dimensional *fastText* vectors pretrained on *wikipedia* (Grave *et al.* 2018),[h] resulting in 400-dimensional vectors. This technique has proven to be very effective for specialized domain corpora (Hazem and Morin 2017; Liu *et al.* 2018). For the general domain corpus, we simply use the *Semeval 2017* given model. Pertaining to the contextualized embedding model, we incorporate the implementation of BERT[i] and ELMo[j] because they both have pretrained models on multiple languages. The BERT implementation has a multilingual model that contains 104 languages, while the ELMo implementation has 44 separate language models. It is worth mentioning that all these models are pretrained on large general corpora (1B words for ELMo and 3.3B for BERT). Finally, we have two types of input embeddings:

- **Static**. The 400-dimensional static word embedding vectors obtained from concatenating the pretrained *fastText* vectors and vectors trained on small specialized domain corpora for the phrase synonymy task and also the 64-dimensional static word embedding vectors provided by the *Semeval 2017* data set for the phrase similarity task.
- **BERT** or **ELMo**. Pretrained contextualized embeddings with feature-based usage setting.

5.2 Bilingual phrase alignment experiments

Data and resources. For the bilingual phrase alignment task, we use the same *Wind Energy* (WE) comparable corpus as in the monolingual tasks. This time we evaluate the English, French, Spanish, and Chinese corpora. Furthermore, we extend the *Breast Cancer* corpus (BC) to an English-Spanish comparable corpus by crawling from a scientific website.[k]

The English BC corpus has 26,716 sentences and the Spanish one has 62,804 sentences. The gold standard was constructed based on the MeSH 2018 thesaurus[l] and contains 108 phrase pairs which exist in our corpus. Concerning the WE corpora, the English, French, Spanish and Chinese parts respectively contain 13,338, 33,887, 29,083 and 17,932 sentences. Hazem and Morin (2016) proposed a reference list consisting of 139 single words for the English-French corpus, while Liu

[d] https://meshb.nlm.nih.gov.
[e] http://alt.qcri.org/semeval2017/task2/index.php?id=data-and-tools.
[f] https://www.cs.york.ac.uk/semeval-2013/task5.
[g] https://deeplearning4j.org.
[h] https://github.com/facebookresearch/fastText/.
[i] https://github.com/huggingface/pytorch-pretrained-BERT.
[j] https://github.com/HIT-SCIR/ELMoForManyLangs.
[k] https://www.sciencedirect.com.
[l] https://meshb.nlm.nih.gov/search.

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

Table 1. Comparison of complexity. n is the input sequence length, d means the model dimension where we assume $d = d_{input} = d_{hidden} = d_{output}$ for simplifying the comparison. k is the kernel width for CNN

	RctNN	CNN	Self-Att	TF-RNN
Computational complexity	$O(n \cdot d^2)$	$O(k \cdot n \cdot d^2)$	$O(n^2 \cdot d)$	$O(\frac{n(n-1)}{2} \cdot d^2)$
Dependency length	$O(n)$	$O(\frac{n}{k})$	$O(1)$	$O(\frac{n}{2})$

Table 2. Detailed term length distribution

Data set-language	1	2	3	4	5	6	Total
BC-en	78	24	5	1	0	0	108
BC-es	96	30	18	4	0	0	148
WE-en	17	62	8	3	0	0	90
WE-fr	17	73	33	8	8	0	139
WE-es	22	73	50	15	1	3	164
WE-zh	34	66	7	3	2	0	112

et al. (2018) provided a gold standard with 73 multiword phrases for the same corpus. Based on the reference list of Liu *et al.* (2018), we propose a new gold standard including also single words. Moreover, we extended this gold standard to other languages while ensuring that all reference lists share the same 90 English phrases to be aligned. Finally, alignment reference lists were obtained for three language pairs: English-French, English-Spanish and English-Chinese. For the sake of comparability, we report results on the data sets of Liu *et al.* (2018) and Hazem and Morin (2016). Table 2 shows the detailed gold term distribution in terms of length for BC and WE corpus. Note that for one English term, we can have multiple correct alignments in the target language.

For the preprocessing and phrase extraction, we also use the *IXA pipes* library to tokenize and lemmatize French and Spanish corpora. It should be noted that the WE Chinese corpus is already pre-segmented. Then we use the *Stanford CoreNLP* library[m] pos-tagger for all languages, then for the phrase extraction we use the same *PKE* tool as mentioned in monolingual tasks. After hapax filtering, each corpus contains roughly 6000 candidate phrases of maximal length 7.

Input cross-lingual embeddings. We implement the bilingual word embedding framework mentioned in Section 2.3 using *deeplearning4j 1.0.0-beta3*.[n] We also use this method to obtain 400-dimensional word embedding vectors as in the monolingual tasks. Recall that this technique follows the idea discussed in Hazem and Morin (2017) and Liu *et al.* (2018). We then apply the bilingual word embedding framework so all word embeddings at input level in each experiment are mapped to a common space. For each language pair, the seed lexicon is selected by a frequency threshold of 50, obtaining around 2000 word pairs. We use unit length normalization, mean centering, matrix whitening, re-weighting, and de-whitening to generate cross-lingual word embeddings.[o] Since our goal is to evaluate the contributions of our system, we will not measure the impact of different pretrained embeddings but prefer to focus on those achieving state-of-the-art results to date.

[m] https://stanfordnlp.github.io/.
[n] https://deeplearning4j.org/.
[o] This is the optimal application order reported by Artetxe *et al.* (2018a).

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

5.3 Training settings

For all our experiments, the dimension of the encoded vector (v_o in Figure 3) for the shared encoder is set to 500. This is also the hidden size for the LSTM decoders. For the *fastText* or ELMo input embeddings, sentences longer than 100 words are cutoff. While for BERT, sentences longer than 150 tokens are cutoff. Because BERT uses *SentencePiece* (Kudo and Richardson 2018) tokenizer and some words are tokenized into several tokens. By truncating sentences, the training is quicker and more stable. We pad the special token [CLS] to the beginning of every sentence for the models with BERT. To extract the features from the BERT model, we sum the output vectors of the last four hidden layers (Devlin *et al*. 2018), this has shown to be the second best method, with only 0.2 F-score point behind concatenating the last four layers which is 4 times less space efficient. The model is trained by a minibatch of 20, which means that given one phrase, we calculate the mean of the cross-entropy loss between 20 predicted and real sentences. We run our experiments for a maximum of 200 epochs with an early-stop condition of three consecutive loss increases. One model with static word embeddings takes about 2 days to train on a single Geforce 1080 Ti GPU with Pytorch 1.0 and Cuda 10 on Ubuntu 16.04, while training with contextualized embeddings takes about 4 days with the feature-based strategy.

5.4 Evaluation settings

The generated phrase vectors are compared by cosine similarity. For the synonymy and bilingual alignment tasks, we simply calculate the cosine of all pre-extracted phrase candidates and rank them. We use the evaluation script provided with the *Semeval2017* data set for the similarity task,[p] and the MAP (Mean Average Precision) score (Manning *et al*. 2008) to evaluate the synonymy and bilingual alignment task:

$$MAP = \frac{1}{|W|} \sum_{1}^{|W|} \frac{1}{Rank_i} \tag{10}$$

where $|W|$ corresponds to the size of the evaluation list, and $Rank_i$ corresponds to the ranking of a correct synonym candidate i.

5.5 Reference methods

Baseline approaches. Regarding the monolingual tasks, we have implemented three types of baseline approaches:

- **Skip-gram-ext**. The extended version of Skip-gram (Artetxe *et al*. 2018b) with 300 dimensional vectors. The implementation is publicly available.[q]
- **Static mean**. The additive approach that has proven to be surprisingly effective (Mikolov *et al*. 2013b; Liu *et al*. 2018).
- **ELMo/BERT mean/reduce/concat**. We extract a single fixed-length vector from the feature-based output sequence of ELMo/BERT with two strategies. The mean is similar to the additive approach where we simply calculate the mean vector over all normalized vectors in the output sequence. The reduce strategy uses one vector to represent the whole sequence: for ELMo it is the last token vector while for BERT we use the output of the hidden layer for the first [CLS] token. The concatenation for ELMo is based on the original ELMo paper (Peters *et al*. 2018) where the authors propose to concatenate the first and the last token to represent a sequence.

[p] https://alt.qcri.org/semeval2017/task1/.
[q] https://github.com/artetxem/phrase2vec.

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

Table 3. Overall MAP comparison for the phrase synonymy task

	Method	Synonymy data set		
		WE-fr	WE-en	BC-en
Baselines	Skip-gram-ext	<0.5	<0.5	23.30
	Static-mean	5.29	12.19	39.65
	BERT-reduce	4.07	10.44	26.04
	BERT-mean	4.49	16.59	36.58
	ELMo-reduce	1.54	4.09	26.23
	ELMo-mean	7.37	5.20	29.27
	ELMo-concat	8.97	9.60	28.28
Context based	Static-CNN	7.42	15.71	35.75
	Static-RecurrentNN	12.89	20.53	42.60
	Static-Transf.	4.62	15.82	35.90
	Static-TF-RNN	**15.06**	**33.47**	**44.84**

For the bilingual alignment task, we have also implemented two baseline classes:

- **Static mean**. This is the same approach as in the monolingual tasks.
- **Co-occurrence based approach**. The compositional approach (Grefenstette 1999; Tanaka 2002; Robitaille *et al*. 2006) is a quick and direct method to align multiword expressions. It is basically a dictionary look-up approach that translates each word via a dictionary and sort all candidates by frequency. Morin and Dailles (2012) proposed a co-occurrence based approach called *compositional approach with context-based method* (CMCBP) to tackle the problem of out of dictionary words. However, this approach can only align phrases of the same length, so we compare only a subset of the multiword phrase pairs.

Encoder–decoder system with other phrase encoders. To compare our proposed TF-RNN and the adapted cross-lingual version, we also implemented several neural networks which do not require structured input: RecurrentNN, CNN, Transformer encoder, and LSTM, which is reported to obtain the best results in Del *et al*. (2018). They all have the same output dimension, and the CNN has a kernel size of 2 and a zero-padding so that even single-word phrases can be encoded. A small Transformer encoder with 4 layers and 4 heads is also implemented, its hidden dimension size of the feed forward is twice the model dimension. Note that it still has many more parameters than other architectures (5 million parameters vs. roughly 0.5 million in other architectures).

6. Results and discussion

6.1 Phrase synonymy and similarity

Overall results on phrase synonymy and similarity tasks are shown in Tables 3 and 4. We compare our proposal with several state-of-the-art methods that can be applied to our tasks.

Our approach with static word embeddings and the TF-RNN as phrase encoder has the best results regarding the phrase synonymy task on specialized domain corpora. The TF-RNN has also

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

Table 4. Overall correlation score comparison for the similarity task. The Semeval correlation score is the harmonic mean of Pearson and Spearman scores. A † indicates that the corresponding corpus for training a neural network is not available. In the case of the context-based approaches, we use the network trained on the Semeval2017 corpus, as it is also a general domain corpus

	Method	Similarity data set	
		Semeval2013†	Semeval2017
Baselines	Skip-gram-ext	0.378	**76.827**
	Static-mean	26.910	38.843
	BERT-reduce	0.754	12.735
	BERT-mean	19.482	36.378
	ELMo-reduce	35.112	37.968
	ELMo-mean	37.991	36.207
	ELMo-concat	36.233	31.420
Context based	Static-CNN	*(29.890)*	42.245
	Static-RecurrentNN	*(21.720)*	42.961
	Static-Transf	***(39.524)***	49.324
	Static-TF-RNN	*(22.003)*	44.382

managed to obtain the third best result for the phrase similarity task on *Semeval2017*. Given that the *Semeval2013* data set does not provide any textual data and the model is trained on the textual corpus of *Semeval2017*, the results on *Semeval2013* for the context prediction approaches are biased by data availability. Moreover, compared to the approaches with contextualized embedding input, the context prediction approaches (last four lines in both tables) have, for the most part, better results on the data sets that provide a textual corpus to train the model. Although the contextualized embedding models capture the inner relation between each component word in a phrase, they cannot exploit the context information of the phrase during the test phase or if the phrase is out of the training corpus. The encoder–decoder training-based approach, however, memorizes and generalizes the context information of different phrases in the training corpus. In addition, if we compare the four phrase encoders from the three unit length normalization, mean centering, matrix whitening context-based approaches, our proposed TF-RNN outperforms the existing neural networks on the synonymy task on every data set and obtains tangible improvements on the similarity task on the *Semeval2017* data set and slightly better results on the *Semeval2013* data set compared to the recurrent neural network. Although the *Transformer* encoder has better results on the similarity task, our encoder has comparable performance while having fewer parameters (0.5M in TF-RNN vs. 5M in *Transformer*). Therefore, we believe that carefully representing the phrase following a relevant syntactical structure can generate better vector representations.

Among the noncontext prediction-based approaches (the first six lines in both tables), first the extended Skip-gram works very poorly for the synonymy task. This is because many phrases during the inference are freely combined so they may not appear in the training corpus. As a consequence, these phrases do not have any representation in the look-up table. This phenomenon can also be observed on the *Semeval2013* similarity task. However, it performs surprisingly well

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

Table 5. Results of our system with the TF-RNN encoder and wrapped context objective with different embeddings

Task	Embeddings		
	ELMo	BERT	Static
WE-fr	9.57	6.47	**15.06**
WE-en	21.39	26.66	**33.47**
BC-en	23.61	26.01	**44.84**
Semeval2013	**24.279**	3.262	22.003
Semeval2017	**47.703**	29.078	44.382

for the similarity task on the *Semeval2017*. The reason probably lies in the fact that *Semeval2017* has a large training corpus and contains all the phrases in our test. We also notice that the contextualized embeddings (from the second to the fifth lines in both tables) are not better than the static embeddings. In fact, the static embeddings hold the best results on the BC and the *Semeval2017* data set. For the contextualized embedding models, it seems that the mean of each output vector better fits our tasks, excepting for the ELMo model on the *Semeval2013* data set. Comparing the BERT and ELMo models with mean representation, the BERT model has relatively respectable results on the English synonymy data sets while the ELMo model is more effective on the French data set and the similarity task. Our explanation is that the BERT model is a multilingual model mixed with 104 languages so it is not surprising that the model is biased by the English training corpus. Conversely, the ELMo French model is a separate model trained only on French data. For the similarity task, the ELMo model largely outperforms the BERT model on the *Semeval2013* data set although the two models have similar results on the *Semeval2017* data set.

In addition to the comparison with other existing methods, we have conducted a series of ablation tests to better understand the behavior of the key components in our system.

Static versus contextualized input embedding results are reported in Table 5.

As stated before, the static embeddings for the synonymy task are open domain pretrained vectors reinforced with specialized domain embeddings, trained on small specialized domain corpora. This solution has been exploited to generate meaningful embedding vectors on specialized domain corpora for bilingual lexicon extraction (Hazem and Morin 2017; Liu *et al*. 2018).

As shown in Table 5, the static embeddings concatenated with specialized domain information achieve clearly better results on the specialized domain data sets (WE and BC). On the contrary, the ELMo model trained on general domain corpora has the best results. We can the deduce that the availability of domain-specific information outweighs the choice of a particular word embedding architecture. For a specialized domain corpus, it is more effective to exploit domain information to improve the model rather than using more advanced architectures or high-coverage open domain resources. Besides, it shows that our system can efficiently incorporate the contextualized embeddings as we obtain the best results on the general domain *Semeval2017* data set, improving the state-of-the art approach by nearly 9 points.

Results with the BERT model are the worst on the French and the *Semeval* data sets. Yet, it outperforms the ELMo model on English synonymy data sets. Again, this confirms that the model is less effective on non-English data sets as we previously discussed. As for the similarity task, we assume that increasing the training size (e.g., 831 phrases in *Semeval2017* vs. 8923 in *WE-en*) would improve the system because the BERT model uses a subword tokenizer that often tokenizes often a word into multiple units. This could make it more difficult to generalize meaningful parameter weights during training.

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

Table 6. Results of our system with the TF-RNN encoder and static embeddings with different training objectives

Task	Training objectives		
	Plain	Context	Wrapped
WE-fr	9.40	13.35	**15.06**
WE-en	30.08	32.85	**33.47**
BC-en	39.48	41.49	**44.84**
Semeval2013	16.759	21.376	**22.003**
Semeval2017	39.223	43.079	**44.382**

Table 7. Comparison of the encoder–decoder framework with a pseudo-siamese network. The two systems use the TF-RNN as phrase encoder

Task	Architecture	
	Pseudo-siamese	Encoder–decoder
WE-fr	3.84	**15.06**
WE-en	11.71	**33.47**
BC-en	32.18	**44.84**
Semeval2013	0.345	**22.003**
Semeval2017	4.164	**44.382**

Wrapped context prediction versus other training objectives results are shown in Table 6.

In order to prove the effectiveness of the proposed training objective, we evaluated two more models using two different training objectives with exactly the same experimental settings. The first one predicts all the sentence tokens, represented by "plain". The second one predicts only the context tokens around the phrase without the wrapped phrase token, represented by "context".

We can clearly see that the wrapped context training objective consistently obtains the best results compared to other possible objectives in our scenario. Although the context prediction strategy is fairly close, adding a wrapped token to replace the phrase allows the system to learn from a syntactically more complete sequence. Predicting all the tokens including the phrase is worse than the context prediction objective even if it predicts a syntactically complete sequence. The reason for this is possibly that predicting the phrase tokens makes the encoder over related to the specific phrase components rather than the generalized features across different but similar phrases, eventually it is difficult for the encoder to generate close vectors for these phrases.

Pseudo-Siamese network versus encoder–decoder system results are shown in Table 7.

There is another applicable framework for training our phrase encoder in an unsupervised way: the pseudo-siamese network (Bromley *et al.* 1993; Zagoruyko and Komodakis 2015; Wang, Li, and Lazebnik 2016). The idea is quite simple, instead of predicting a sequence of context tokens from a decoder, the network minimizes the vector distance between the phrase and the context vector. We use an LSTM to encode the context with a self-attention mechanism (Lin *et al.* 2017).

We can see that the pseudo-siamese network performs very poorly on all data sets, with extremely large drops on the similarity. It is somewhat unexpected for us since both training

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

Table 8. Overall MAP for all phrase alignment. *sw*, *n2n*, and *p2q*, respectively, mean single-word to single-word, same length multiword, and variable length phrase alignment. We do not present the results of the distributional approach on the English-Chinese corpus because we do not have enough resources to build the co-occurrence matrix as in the other language pairs because the distributional approach requires a high coverage bilingual dictionary. Furthermore, if the dictionary does not use the same Chinese word segmentation approach as the WE corpus, it is even harder to find the words. † means the original data set used in Liu *et al.* (2018)

Data set		Method		Encoder				Our
Corpus	Phrases	CMCBP	ADD	Rec.	CNN	LSTM	TRAN.	method
	sw (72)	35.72	47.46	46.71	45.12	46.25	43.37	**47.76**
BC	n2n (21)	68.73	81.10	28.52	62.10	50.05	59.26	**86.11**
en-es	p2q (9)	–	42.18	1.11	10.65	7.04	4.49	**49.11**
	all (108)	–	52.85	36.78	43.04	43.72	43.22	**55.40**
	sw (15)	65.56	78.25	77.22	78.33	79.36	**85.56**	79.44
WE	n2n (61)	42.09	57.37	6.16	40.84	18.64	41.82	**62.19**
en-fr	p2q (14)	–	15.83	<0.5	10.07	9.09	12.35	**37.95**
	all (90)	–	55.77	17.25	43.33	27.42	44.53	**62.10**
	sw (15)	63.35	77.92	**88.89**	75.78	87.18	84.44	87.62
WE	n2n (61)	35.94	**62.68**	7.31	40.33	23.07	44.68	61.35
en-es	p2q (14)	–	43.28	<0.5	28.57	17.86	37.20	**46.21**
	all (90)	–	62.20	19.77	44.41	32.94	50.14	**63.38**
	sw (17)	–	53.43	70.26	**76.47**	71.43	65.92	66.50
WE	n2n (47)	–	23.34	17.53	16.55	**25.24**	18.86	23.01
en-zh	p2q (26)	–	4.97	5.13	7.60	2.37	5.80	**12.32**
	all (90)	–	22.67	23.91	25.28	27.36	23.98	**28.13**
WE	n2n (40)	67.32	78.36	46.07	68.51	44.82	48.47	**88.01**
en-fr	p2q (33)	–	34.38	2.38	20.01	7.93	28.25	**41.83**
Liu2018†	all (73)	–	58.48	26.06	46.59	28.13	39.33	**67.13**

approaches are inspired by the Harris distributional hypothesis (Harris 1954). This may be due to the nature of the comparison tasks or the small size of our training samples. It perhaps explains why encoder–decoder systems are becoming more popular in recent studies compared to others.

6.2 Bilingual phrase alignment

Table 8 shows the overall results on all test phrases. Since the distributional approach (Morin and Daille 2012) does not include the alignment of variable length phrases, we ignore the corresponding results in the table.

It is clearly shown that the proposed method has a better overall performance. Especially when it comes to different length phrase alignments, the new approach significantly improves the MAP with an average score of 8.8 points. This proves that the proposed method is able to produce high-quality alignment for phrases of variable length. Keep in mind that the different length

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

Table 9. MAP % for single-word phrase alignment

	BC	WE			
Method	en-es	en-fr	en-es	en-zh	en-fr-HM
Mikolov *et al*. (2013a)	39.96	91.33	87.27	45.88	79.47
Artetxe *et al*. (2018a)	**49.13**	**95.56**	**90.39**	**73.52**	**84.01**
Our method	45.96	89.44	88.89	58.75	82.23

distribution represents a small proportion of all test phrases except for the English-Chinese corpus, so the overall score would be furthermore improved if we had uniform distribution for all kinds of alignment. The second best method is the previously described addition approach, which obtains good results (Mikolov *et al*. 2013b; Del *et al*. 2018). However, we observe that between linguistically distant language pairs (English-Chinese), all encoder–decoder systems outperform the addition-based approach. The CNN has some interesting results in same length alignment, and the LSTM is powerful concerning short phrase alignment but unlike in Del *et al*. (2018), it falls behind on other types. This difference may be explained by the fact that they limit the alignment to two-word phrases.

The transformer encoder does not obtain better results than the addition-based approach nor better results than the other encoders. First, for all the addition is still more adaptive and effective for short sequence comparison between linguistically close language pairs (Hazem and Morin 2017; Liu *et al*. 2018; Del *et al*. 2018). Second, as we set a maximum epoch of 200, we think that the transformer encoder may not be converged after 200 epochs because it has a much bigger parameter-sample ratio than the other encoders. Finally, transformer architectures are basically multihead self-attentions, which are designed for capturing the relations in long sequences while we encode mostly short sequences.

The relatively poor results on the English-Chinese corpus may be due to the segmentation of Chinese words. More concretely, as the input vectors for the Chinese sequences are at word-level, many words in our gold standard are not segmented in the same way as in the given corpus which is already pre-segmented. We would like to replace the word-level embeddings by character-level ones in our future works.

Concerning the single-word alignment on BC, 25 among the 72 single words are in fact acronyms which are particularly difficult to align. This would explain why the single-word alignment has much poorer results than other distributions. The proposed method obtains strong results for single-word alignment and we believe this happens because the system sees more single-word alignment samples generated by the pseudo back-translation during training.

In order to show that the proposed method can still maintain a reasonable performance on single words, we present in Table 9 the results for single words compared to state-of-the-art work on bilingual word embedding (Artetxe *et al*. 2018a), including the 139 English-French single word data set of Hazem and Morin (2016) (suffixed -HM in the Table 9). To be comparable, we only test on single-word phrases and the candidate list is limited to all single words in the corpus vocabulary. In our data sets, the source English words are the same 15 as in the *sw* line of Table 8.

We can see that in general, compared to Artetxe *et al*. (2018a), the proposed approach does not significantly degrade the results except for the English-Chinese words. In addition, we succeed in maintaining a better result with regard to the original transformation matrix method (Mikolov *et al*. 2013a) with only one exception on the English-French *wind energy* data set. This shows that our approach is not biased by the compositionality of the multiword expressions.

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

Table 10. Alignment examples within top 2 candidates ("␣" is the segmentation point for Chinese words)

Data set	Source	Addition	Our method
BC	breast cancer	cáncer mamario	cáncer de mama
en-es	cell death	muerte celular	muerte
WE	blade tip	angle des pales	côté supérieur de la pale
en-fr	Darrieus rotor	rotor tripale	rotor vertical
WE	airflow	freno aerodinámico	flujo de aire
en-es	wind power plant	electricidad del viento	planta eólica
WE	wind vane	偏航␣电机	风向标
en-zh	electricity power	电力	电力

6.3 Bilingual phrase alignment qualitative analysis

For a better understanding of how the proposed method succeeds or fails to align different types of phrases, we analyzed some of the alignments proposed by our system.

Table 10 shows examples extracted from the top 2 nearest candidates to the source phrase in column 2. Again we see that the proposed method is capable of generating better results over different types of alignment. In the first example, with the proposed approach, the source phrase *breast cancer* is aligned to *cáncer de mama* (lit. "cancer of breast"), which is the expected phrase in Spanish and is far more idiomatic than *cáncer mamario* (lit. "cancer mammary") obtained by the addition approach. In line 7, we see that the perfect translation for *wind vane* is found by our proposal: 风向标, while the additive approach finds 偏航␣电机 (lit. "yaw electric machine"). Besides, examples in lines 3, 5, 6, 7, and 8 are all composed of phrases of variable length, and the corresponding reference phrase can be found in the fourth column. Interestingly, we find that the proposed system find paraphrases referring to fairly domain-specific phrases like *blade tip,* which is aligned to *côté supérieur de la pale* (lit. "side top of the blade"). This is also the case for *Darrieus rotor* aligned to *rotor vertical*, which is an outstanding result since the Darrieus rotor is a kind of vertical rotor.

Though the proposed method performs generally well on phrases, we observe that it occasionally over emphasizes occasionally too much the syntactic head in a multiword phrase. For instance, in the second example, *cell death* is aligned to *muerte* ("death"), while the addition-based approach manages to align it to *muerte celular* (lit. "death cellular"), which is the reference phrase in Spanish. Undoubtedly, *death* is the syntactic head for the noun phrase *cell death*, it is clear that the proposed method puts more weight on the syntactic information rather than the compositional property for this phrase. In a more generalized manner, the translations for English source phrases with syntactical patterns such as *ADJ NOUN* will be only *NOUN*. This also explains why we do not obtain better results on equal-length phrase alignment on the English-Spanish and English-Chinese *wind energy* corpora (Table 8). This bias could be due to the increased amount of single-word phrase samples of the pseudo back-translation reinforced learning. This suggests that we could possibly improve the system by adding synthetic translations for multiword phrases during the training.

7. Conclusion and perspective

Significant advances have been achieved in bilingual word-level alignment, yet the challenge remains for phrase-level alignment. Moreover, the need for parallel data is a critical drawback for the alignment task. This work proposes a system that alleviates these two problems: a unified

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

phrase representation model using cross-lingual word embeddings as input and an unsupervised training algorithm inspired by recent works on neural machine translation.

The proposed system consists in a encoder–decoder system where for the encoder part we introduce a new short sequence encoder called a *tree-free recursive neural network* (TF-RNN), that constructs cross-lingual representations of phrases of any length and takes into account word order. For the decoder part, we use a two-layer LSTM that decodes these representations w.r.t their contexts. As for the training strategy, in order to train the network in an unsupervised way, we also incorporate a pseudo back translation mechanism. Experiments on five data sets show that the adaptability that our method offers does not imply performance drawbacks. In fact, on the bilingual phrase alignment task results are on par with the state of the art. As for the alignment of phrases of different lengths, our method improves the latest results by a mean of 8.8 points in MAP and seems mainly limited by segmentation issues, which we intend to address in future works using character-level embeddings.

We would also like to continue studying more extensive evaluations in our future work and further study the extract-edit approach (Wu *et al*. 2019) to improve our system. Based on a method similar to back-translation, we could use extracted and edited phrases as the synthetic translations which would avoid any misleading caused by the poor translations. Finally, two strategies could be more deeply explored regarding data selection: pretraining corpus merge and post-training embedding merge. The former investigates the nature and the quality of the corpora and trains the word embeddings with one finely merged corpus. The latter trains separately word embeddings from the general and specialized domain corpora and then merges these word embeddings. We would like to study the behavior of using different merging approaches such as a specific layer related to the merge or multitask learning for both separated embeddings.

References

Agerri R., **Bermudez J.** and **Rigau G.** (2014). Ixa pipeline: Efficient and ready to use multilingual nlp tools. In Chair N.C.C., Choukri K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J. and Piperidis, S. (eds), *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Artetxe M., **Labaka G.** and **Agirre E.** (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 (EMNLP'16)*, Austin, TX, USA, pp. 2289–2294.

Artetxe M., **Labaka G.** and **Agirre E.** (2018a). Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI'18)*, New Orleans, LA, USA, pp. 5012–5019.

Artetxe M., **Labaka G.** and **Agirre E.** (2018b). Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*, Brussels, Belgium, pp. 3632–3642.

Artetxe M., **Labaka G.**, **Agirre E.** and **Cho K.** (2018c). Unsupervised neural machine translation. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*, Vancouver, Canada.

Bahdanau D., **Cho K.** and **Bengio Y.** (2014). Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473.

Bromley J., **Guyon I.**, **LeCun Y.**, **Säckinger E.** and **Shah R.** (1993). Signature verification using a "siames" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence* **7**(4), 669–688.

Bullinaria J.A. and **Levy J.P.** (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods* **39**(3), 510–526.

Camacho-Collados J., **Pilehvar M.T.**, **Collier N.** and **Navigli R.** (2017). SemEval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada. Association for Computational Linguistics, pp. 15–26.

Chen Y., **Liu Y.**, **Cheng Y.** and **Li V.O.** (2017). A teacher-student framework for zero-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*, Vancouver, Canada, pp. 1925–1935.

Cho K., **van Merrienboer B.**, **Gulcehre C.**, **Bahdanau D.**, **Bougares F.**, **Schwenk H.** and **Bengio Y.** (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, Doha, Qatar, pp. 1724–1734.

Church K.W. and **Hanks P.** (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics* **16**(1), 22–29.

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

Dagan I., **Pereira F.** and **Lee L.** (1994). Similarity-based estimation of word cooccurrence probabilities. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics (ACL '94)*, Stroudsburg, PA, USA, pp. 272–278.

Del M., **Tättar A.** and **Fishel M.** (2018). Phrase-based unsupervised machine translation with compositional phrase embeddings. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, Belgium, Brussels, pp. 361–367.

Devlin J., **Chang M.-W.**, **Lee K.** and **Toutanova K.** (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805.

Elman J.L. (1990). Finding structure in time. *Cognitive Science* **14**(2), 179–211.

Faruqui M. and **Dyer C.** (2014). Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL'14)*, Gothenburg, Sweden, pp. 462–471.

Firat O., **Sankaran B.**, **Al-Onaizan Y.**, **Yarman Vural F.T.** and **Cho K.** (2016). Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*, Austin, TeX, USA, pp. 268–277.

Fung P. (1995). Compiling bilingual lexicon entries from a non-parallel english-chinese corpus. In *Proceedings of the 3rd Annual Workshop on Very Large Corpora (VLC'95)*, Cambridge, MA, USA, pp. 173–183.

Garten J., **Sagae K.**, **Ustun V.** and **Dehghani M.** (2015). Combining distributed vector representations for words. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, Denver, CO, USA, pp. 95–101.

Gehring J., **Auli M.**, **Grangier D.**, **Yarats D.** and **Dauphin Y.N.** (2017). Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML'17)*, Sydney, Australia, pp. 1243–1252.

Goikoetxea J., **Agirre E.**, and **Soroa A.** (2016). Single or multiple? combining word representations independently learned from text and wordnet. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*, Phoenix, AZ, USA, pp. 2608–2614.

Goller C. and **Küchler A.** (1996). Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, Washington, DC, USA, pp. 347–352.

Grave E., **Bojanowski P.**, **Gupta P.**, **Joulin A.** and **Mikolov T.** (2018). Learning word vectors for 157 languages. In *Proceedings of 11th Edition of the Language Resources and Evaluation Conference (LREC'18)*, Miyazaki, Japan, pp. 3483–3487.

Grefenstette G. (1999). The world wide web as a resource for example-based machine translation tasks. In *Proceedings of the ASLIB Conference on Translating and the Computer 21*, London, UK.

Harris Z. (1954). Distributional structure. *Word* **10**(2–3), 146–162.

Hazem A. and **Daille B.** (2018). Word embedding approach for synonym extraction of multi-word terms. In *Proceedings of the 11th edition of the Language Resources and Evaluation Conference (LREC'18)*, Miyazaki, Japan, pp. 297–303.

Hazem A. and **Morin E.** (2016). Efficient data selection for bilingual terminology extraction from comparable corpora. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING'16)*, Osaka, Japan, pp. 3401–3411.

Hazem A. and **Morin E.** (2017). Bilingual word embeddings for bilingual terminology extraction from specialized comparable corpora. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP'17)*, Taipei, Taiwan, pp. 685–693.

He D., **Xia Y.**, **Qin T.**, **Wang L.**, **Yu N.**, **Liu T.-Y.** and **Ma W.-Y.** (2016). Dual learning for machine translation. In *Advances in Neural Information Processing Systems 29 (NIPS'16)*, pp. 820–828.

Huang J., **Cai X.** and **Church K.** (2020). Improving bilingual lexicon induction for low frequency words. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics, pp. 1310–1314.

Irsoy O. and **Cardie C.** (2014). Deep recursive neural networks for compositionality in language. In *Advances in Neural Information Processing Systems 27 (NIPS'14)*, pp. 2096–2104.

Johnson M., **Schuster M.**, **Le Q.V.**, **Krikun M.**, **Wu Y.**, **Chen Z.**, **Thorat N.**, **Viégas F.**, **Wattenberg M.**, **Corrado G.**, **Hughes M.** and **Dean J.** (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics* **5**, 339–351.

Korkontzelos I., **Zesch T.**, **Zanzotto F.M.** and **Biemann C.** (2013). Semeval-2013 task 5: Evaluating phrasal semantics. In *Second Joint Conference on Lexical and Computational Semantics (SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, Atlanta, Georgia, USA. Association for Computational Linguistics, pp. 39–47.

Kudo T. and **Richardson J.** (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Brussels, Belgium. Association for Computational Linguistics, pp. 66–71.

Lample G. and **Conneau A.** (2019). Cross-lingual language model pretraining. CoRR, abs/1901.07291.

Lample G., **Conneau A.**, **Denoyer L.** and **Ranzato M.** (2018). Unsupervised machine translation using monolingual corpora only. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*, Vancouver, Canada.

Laville M., **Hazem A.**, **Morin E.** and **Langlais P.** (2020). Data selection for bilingual lexicon induction from specialized comparable corpora. In *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online). International Committee on Computational Linguistics, pp. 6002–6012.

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

Lazaridou A., **Dinu G.** and **Baroni M.** (2015). Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP'15)*, Beijing, China, pp. 270–280.

Le P. and **Zuidema W.** (2014). The inside-outside recursive neural network model for dependency parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, Doha, Qatar, pp. 729–739.

Le P. and **Zuidema W.** (2015). Compositional distributional semantics with long short term memory. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, Denver, CO, USA, pp. 10–19.

Lee J., **Cho K.** and **Hofmann T.** (2017). Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics* **5**, 365–378.

Lin Z., **Feng M.**, **dos Santos C.N.**, **Yu M.**, **Xiang B.**, **Zhou B**. and **Bengio Y**. (2017). A structured self-attentive sentence embedding. CoRR, abs/1703.03130.

Liu J., **Morin E.** and **Peña Saldarriaga S.** (2018). Towards a unified framework for bilingual terminology extraction of single-word and multi-word terms. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING'18)*, Santa Fe, NM, USA, pp. 2855–2866.

Liu J., **Morin E.**, **Peña Saldarriaga S.** and **Lark J.** (2020). A unified and unsupervised framework for bilingual phrase alignment on specialized comparable corpora. In *24th European Conference on Artificial Intelligence (ECAI)*, Santiago de Compostela, Spain.

Luong M.-T., **Pham H.** and **Manning C.D.** (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*, Lisbon, Portugal, pp. 1412–1421.

Manning C.D., **Raghavan P.** and **Schütze H.** (2008). *An Introduction to Information Retrieval*. New York: Cambridge University Press.

Mikolov T., **Le Q.V.** and **Sutskever I.** (2013a). Exploiting similarities among languages for machine translation. CoRR, abs/1309.4168.

Mikolov T., **Sutskever I.**, **Chen K.**, **Corrado G.S.** and **Dean J.** (2013b). Distributed representations of words and phrases and their compositionality. In *Advances Neural Information Processing Systems 26 (NIPS'13)*, pp. 3111–3119.

Mitchell J. and **Lapata M.** (2009). Language models based on semantic composition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP'09)*, Singapore, pp. 430–439.

Morin E. and **Daille B.** (2012). Revising the compositional method for terminology acquisition from comparable corpora. In *Proceedings of the 24rd International Conference on Computational Linguistics (COLING'12)*, Mumbai, India, pp. 1797–1810.

Niwa Y. and **Nitta Y.** (1994). Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In *Proceedings of the 15th Conference on Computational Linguistics (COLING'94)*, Kyoto, Japan, pp. 304–309.

Paulus R., **Socher R.** and **Manning C.D.** (2014). Global belief recursive neural networks. In *Advances in Neural Information Processing Systems 27 (NIPS'14)*, pp. 2888–2896.

Peng X., **Lin C.** and **Stevenson M.** (2021). Cross-lingual word embedding refinement by ℓ_1 norm optimisation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online. Association for Computational Linguistics, pp. 2690–2701.

Pennington J., **Socher R.** and **Manning C.** (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, Doha, Qatar, pp. 1532–1543.

Peters M., **Neumann M.**, **Iyyer M.**, **Gardner M.**, **Clark C.**, **Lee K.** and **Zettlemoyer L.** (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'18)*, New Orleans, LA, USA, pp. 2227–2237.

Radford A., **Wu J.**, **Child R.**, **Luan D.**, **Amodei D.** and **Sutskever I.** (2019). Language models are unsupervised multitask learners. https://github.com/openai/gpt-2.

Rajpurkar P., **Zhang J.**, **Lopyrev K.** and **Liang P.** (2016). Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*, Austin, TX, USA, pp. 2383–2392.

Rapp R. (1999). Automatic identification of word translations from unrelated english and german corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, College Park, Maryland, USA, pp. 519–526.

Robitaille X., **Sasaki Y.**, **Tonoike M.**, **Sato S.** and **Utsuro T.** (2006). Compiling french-japanese terminologies from the web. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'06)*, Trento, Italy, pp. 225–232.

Saha A., **Khapra M.M.**, **Chandar S.**, **Rajendran J.** and **Cho K.** (2016). A correlational encoder decoder architecture for pivot based sequence generation. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING'16)*, Osaka, Japan, pp. 109–118.

Sennrich R., **Haddow B.** and **Birch A.** (2016). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL'16)*, Berlin, Germany, pp. 86–96.

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press

Shigeto Y., **Suzuki I.**, **Hara K.**, **Shimbo M.** and **Matsumoto Y.** (2015). Ridge regression, hubness, and zero-shot learning. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD'15)*, Porto, Portugal, pp. 135–151.

Smith S.L., **Turban D.H.P.**, **Hamblin S.** and **Hammerla N.Y.** (2017). Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17)*, Toulon, France.

Socher R., **Bauer J.**, **Manning C.D.** and **Andrew Y. N.** (2013a). Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL'13)*, Sofia, Bulgaria, pp. 455–465.

Socher R., **Manning C.D.** and **Ng A.Y.** (2010). Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, Vancouver, Canada, pp. 1–9.

Socher R., **Perelygin A.**, **Wu J.**, **Chuang J.**, **Manning C.D.**, **Ng A.** and **Potts C.** (2013b). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*, Seattle, WA, USA, pp. 1631–1642.

Sutskever I., **Vinyals O.** and **Le Q.V.** (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27 (NIPS'14)*, pp. 3104–3112.

Tanaka T. (2002). Measuring the similarity between compound nouns in different languages using non-parallel corpora. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02)*, Stroudsburg, PA, USA, pp. 1–7.

Turney P.D. and **Pantel P.** (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research* **37**(1), 141–188.

Vaswani A., **Shazeer N.**, **Parmar N.**, **Uszkoreit J.**, **Jones L.**, **Gomez A.N.**, **Kaiser L.u.** and **Polosukhin I**. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NIPS'17)*, pp. 5998–6008.

Vincent P., **Larochelle H.**, **Bengio Y.** and **Manzagol P.-A.** (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning (ICML'08)*, New York, NY, USA, pp. 1096–1103.

Wang A., **Singh A.**, **Michael J.**, **Hill F.**, **Levy O.** and **Bowman S.** (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Brussels, Belgium, pp. 353–355.

Wang L., **Li Y.** and **Lazebnik S.** (2016). Learning deep structure-preserving image-text embeddings. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*, Las Vegas, NV, USA, pp. 5005–5013.

Williams A., **Nangia N.** and **Bowman S.** (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 (ACL'18)*, New Orleans, LA, USA, pp. 1112–1122.

Wu J., **Wang X.** and **Wang W.Y.** (2019). Extract and edit: An alternative to back-translation for unsupervised neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'19)*, Minneapolis, Minnesota, pp. 1173–1183.

Xing C., **Wang D.**, **Liu C.** and **Lin Y.** (2015). Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'15)*, Denver, CO, USA, pp. 1006–1011.

Yang Z., **Chen W.**, **Wang F.** and **Xu B.** (2018). Unsupervised neural machine translation with weight sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'18)*, Melbourne, Australia, pp. 46–55.

Zagoruyko S. and **Komodakis N.** (2015). Learning to compare image patches via convolutional neural networks. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*, Boston, MA, USA, pp. 885–894.

Zellers R., **Bisk Y.**, **Schwartz R.** and **Choi Y.** (2018). Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*, Brussels, Belgium, pp. 93–104.

Zhang J. and **Zong C.** (2016). Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*, Austin, TX, USA, pp. 1535–1545.

Zhang Y., **Gaddy D.**, **Barzilay R.** and **Jaakkola T.** (2016). Ten pairs to tag – multilingual pos tagging via coarse mapping between embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'16)*, San Diego, CA, USA, pp. 1307–1317.

Cite this article: Liu J, Morin E, Peña Saldarriaga S and Lark J (2023). From unified phrase representation to bilingual phrase alignment in an unsupervised manner. *Natural Language Engineering* **29**, 643–668. https://doi.org/10.1017/S1351324922000328

https://doi.org/10.1017/S1351324922000328 Published online by Cambridge University Press