Unsupervised Learning of Cross-Lingual Symbol Embeddings Without Parallel Data

Mark Granroth-Wilding
University of Helsinki
mark.granroth-wilding@helsinki.fi

Hannu Toivonen
University of Helsinki
hannu.toivonen@cs.helsinki.fi

Abstract

We present a new method for unsupervised learning of multilingual symbol (e.g. character) embeddings, without any parallel data or prior knowledge about correspondences between languages. It is able to exploit similarities across languages between the distributions over symbols' contexts of use within their language, even in the absence of any symbols in common to the two languages. In experiments with an artificially corrupted text corpus, we show that the method can retrieve character correspondences obscured by noise. We then present encouraging results of applying the method to real linguistic data, including for low-resourced languages. The learned representations open the possibility of fully unsupervised comparative studies of text or speech corpora in low-resourced languages with no prior knowledge regarding their symbol sets.

1 Introduction

Linguistic typology aims to map connections and similarities between different languages or dialects along multiple dimensions of comparison. A large proportion of languages spoken today have few speakers and little data annotated with linguistic analyses such as syntactic parses or part-of-speech tags. This makes mapping their typology difficult, but doing so could help in developing just such resources, for example by language transfer. There may exist digital text in these languages (e.g. forum posts or newspapers), or field recordings of speech. We attempt to learn about a language's typology purely from its surface form.

We focus on languages known to be fairly closely related (e.g. in the same language family), but where knowing more about the precise nature of the typology (e.g. regular sound correspondences in cognate words or differences in morphology) could help with resource development. One example is the Uralic family, which contains many low-resourced languages and dialects.

To compare languages' surface forms, we must first address how to compare their basic units, characters in the case of text (List, 2014). Even closely related languages may use different writing systems, conventions, or transcription practices, as well as having systematic linguistic differences. These considerations mean that, without prior knowledge of a correspondence between two languages, it may not make sense to assume that, say, the letter *a* in one is directly comparable to *a* in the other. For example, Swedish *å* typically corresponds Finnish *o*, and loanwords from Swedish to Finnish replace the former with the latter. Whilst such direct and well known correspondences can easily be written down by someone familiar with the language pair, capturing less clear-cut or systematic correspondences, and doing so for a large number of low-resourced language pairs, is labour intensive.

In an extreme case, two corpora may use completely distinct symbol sets, e.g. different scripts. There may be systematic linguistic differences that create a close correspondence between different symbols across languages (List, 2014), such as the phonological correspondence between Frisian *f* and Danish *v* (Fenna et al., 2014). It may also be desirable to find correspondences between sequences of symbols, e.g. Spanish *ñ* and Portuguese *nh*.

We tackle this problem using unsupervised learning of vector representations (embeddings) of symbols, learning purely from unannotated, unaligned linguistic corpora. Here, we apply our method to text, learning representations of characters, but it is equally applicable to other sequences, such as phonetic sequences from speech. To be applicable to extreme cases of very little overlap between symbol vocabularies (e.g. different scripts,

Proceedings of the Society for Computation in Linguistics (SCiL) 2019, pages 19-28.
New York City, New York, January 3-6, 2019

or types of phonological transcription), it does not assume a correspondence even between common symbols. E.g., if both use *a*, it treats *a* in the two languages as distinct symbols (*1:a* and *2:a*). This means that, where such correspondences *are* found, we know that they are motivated by statistical regularities in their usages, rather than any initial bias. It may learn that *1:a* corresponds to *2:a*, or to *2:ä*, or that it has a weak correspondence to multiple characters. This makes for a challenging learning task, since it becomes impossible to exploit the idea behind typical distributional methods – that similar symbols can be recognized by similarities between their contexts of occurrences – since the contexts across languages consist of symbols from distinct sets.

We present a method that is able to discover similarities between inter-lingual symbol pairs by exploiting similarities between their respective *intra*-lingual distributions over contexts of occurrence. It must recognize that *1:a* plays a role in relation to *other symbols in language 1* that is similar to, say, *2:ä*'s role in relation to *other symbols in language 2*. It does not rely on parallel or comparable corpora, so is robust to use on whatever corpora are available for the languages of interest.

In this paper, we describe our learning method, XSYM (§3). Then we present two sets of experiments. In the first (§4), we use artificially corrupted linguistic data, allowing us to observe how well the technique recovers known mappings between character pairs obscured by the corruption. In the second (§5), we demonstrate encouraging initial results of applying the method to real linguistic data, including several low-resourced pairs, which show that it is able to build a coherent space of characters, for example placing the majority of identical characters in two related languages close to each other. This demonstrates its potential to recover correspondences between symbol pairs on the basis of distributional statistics without any other connection between the observed corpora.

Code for data preprocessing and model training, as well as trained embeddings, are available online[1].

2 Related work

Like us, Tsvetkov et al. (2016) employ a language modeling objective with neural networks to learn multilingual embeddings for symbols (phones). They supply typological information to improve the representations. We believe that the present method is better suited to direct cross-lingual comparison of symbols and, since we aim to discover typological information, do not incorporate this in the input. Östling and Tiedemann (2016) use a character-level, multilingual language model to learn vectors to represent languages. Whilst their model shares information between languages, we focus on modeling commonalities at the level of symbol embeddings. We expect the cross-lingual information our method captures to be complementary to that in the language vectors.

A particular area where symbol alignment is required is *cognate discovery* – finding words with a common linguistic origin. List (2014) describes uses of string alignment methods, the predominant approach in the literature. He distinguishes *paradigmatic* aspects (correspondences between basic units, like phones) and *syntagmatic* aspects (comparisons in terms of sequence structure). Approaches to paradigmatic modeling include: assuming a simple set of correspondences between symbols, e.g. aligning identical symbols (Brew et al., 1996; Kondrak, 2000; Prokić et al., 2009); abstracting or normalizing symbols to comparable classes (Kondrak and Hirst, 2002; Diana Inkpen, 2005; List, 2012); and learning scoring functions or mappings to align symbols, often initializing using one of the previous assumptions (Pirkola et al., 2003; Mulloni and Pekar, 2006; Mulloni, 2007; Kondrak, 2009; Delmestri and Cristianini, 2010; Gomes and Lopes, 2011; Ciobanu and Dinu, 2014). Our approach in these terms is to learn paradigmatic correspondences from purely syntagmatic information. Some methods handle sound (e.g. phone) sequences, others text: ours, like Tsvetkov et al. (2016), can be applied to either. In contrast to alignment approaches, Hall and Klein (2010) use a Bayesian model of language change to account for differences in phonetic surface forms. McCoy and Frank (2018) use context-based character embeddings for cognate discovery and propose a method to discover cognates in a low-resourced language via a better-resourced pivot language. Our embeddings could be used with the same cognate alignment technique and evaluation scheme in future work. Our method provides an alternative, potentially more flexible, way to align with a low-resourced language.

[1] https://mark.granroth-wilding.co.uk/papers/unsup_symbol/

Most methods depend to some degree on linguistic resources. Many require a list of known cognate pairs (Mulloni and Pekar, 2006; Mulloni, 2007; Delmestri and Cristianini, 2010; Gomes and Lopes, 2011; Ciobanu and Dinu, 2014), or a manually aligned corpus (Navlea and Todirascu, 2011; List, 2012), others language-specific knowledge about symbols (Kondrak, 2000) or NLP tools, such as part-of-speech taggers (Brew et al., 1996; Navlea and Todirascu, 2011). Hall and Klein (2010) require a phylogeny of the input languages. We avoid reliance on any language-specific resources.

The issue of cross-lingual symbol alignment also arises in other tasks and similar approaches are used. For example, methods for computing language similarity from the surface form fall into the same categories described above for cognate identification (Batagelj et al., 1992; Kita, 1999; Petroni and Serva, 2008; Gamallo et al., 2017).

Unsupervised or semi-supervised learning of multilingual representations has been addressed at other levels of analysis (e.g. Kuhn, 2004; Snyder et al., 2009; Christodoulopoulos et al., 2012). Many could be applied to unsupervised typology, since linguistic typology concerns all levels of analysis, so are complementary to that we present. Conneau et al. (2017) present unsupervised learning of multilingual word embeddings. This could be applied to low-resourced languages and combined with our method to identify words that are related in both etymology and meaning (the *Specific Homologue Detection Problem*, List, 2014).

Conneau et al.'s learning problem is similar to ours, applied to word meaning rather than symbol correspondence. Whilst a similar technique could perhaps be applied to the present task, our method focuses specifically on similarities in local contexts of symbol use, rather than similarities in the structure of embedding spaces, which are less informative in the case of small vocabularies of characters or phonemes.

3 Method

We describe a model that assigns language model-type scores to short sequences of symbols. We train the model and use the learned embeddings and n-gram composition function. We are not ultimately interested in the predictive model, only the derived representations. The learning technique follows other representation learning algorithms (such as Mikolov et al., 2013) in using *negative sampling*. However, these methods cannot be applied directly, since the fact that the vocabularies of observed contexts are distinct for the two languages means they are unable to discover similarities between characters across languages.

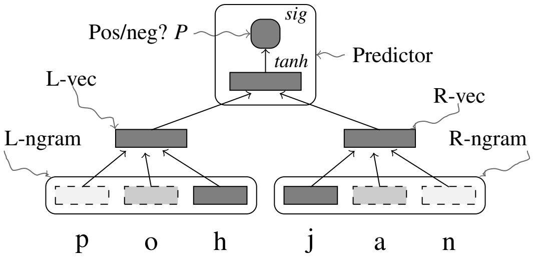

Figure 1: Structure of the neural network used to learn cross-lingual embeddings. The embeddings are used in the bottom layer. The output is a value between 0 and 1 that is used in the BPR objective function, with either positive or negative examples provided at the inputs.

In each sample seen at training time (*pohjan* in Fig. 1), all characters are from the same language, so the vectors for a Finnish character are affected only by other Finnish characters surrounding it. It is therefore possible that the resulting embeddings are grouped by language, effectively learning an independent predictor for each language. Training a high-capacity model (like an RNN) on multilingual data tends to result in this outcome. However, limiting the capacity of the network can force the model to share information between the languages at the level of embeddings. It then benefits the model to learn embeddings that exploit similarities across languages in the relationships between adjacent character sequences within a language. For example, if *a* is often followed by *b* in both languages, and there are also similarities between usage of *1:b* and *2:b*, the model can exploit this by learning similar vectors for *1:a* and *2:a*, and simultaneously *1:b* and *2:b*.

3.1 Model

Our unsupervised representation learning method, XSYM, consists of a feedforward neural network (Fig. 1) that takes as input a short sequence of characters and predicts whether or not it is a real sample from one of the languages in the training data. The character vocabularies are distinguished in the input: e.g. *fi:a* is distinct from *et:a*. The required limitation of capacity mentioned above is achieved by limiting the size of the layers and using only a small number of layers for the predictor.

The length of the input sequence is variable. Each side (L-ngram and R-ngram) may be a single symbol, represented by the symbol's embedding (which becomes L-vec/R-vec), or a bi- or tri-gram, whose embeddings are concatenated and projected by a linear transformation to get a vector for the n-gram, L-vec or R-vec. Separate transformations are learned for bi-grams and tri-grams. The same embeddings are used in each input position and the same composition function on both sides. L-vec and R-vec have the same size as the embeddings learned for individual characters.

The outputs of the two compositions are passed to a predictor function: a tanh layer and a sigmoid activation for the final output node, P. Varying the size of the two ngrams (L- and R-ngram) independently, so that a bigram is sometimes observed beside a unigram, sometimes a bigram, etc, causes the composed representations to reside in the same vector space, since they are inputs to the same predictor function.

In the experiments, we use an embedding (and composed n-gram representation) size of 30. The hidden layer in the predictor also has 30 nodes.

3.2 Learning

Positive samples are taken by passing a sliding window over the text, alternating corpora. Each positive sample is accompanied by a randomly generated negative. The positive and negative output values are used with a *Bayesian Personalized Rank* (BPR) objective function for training. BPR has been successfully used for similar representation learning tasks, where negative data is not directly available: it encourages negative samples to be ranked lower than corresponding positives (Riedel et al., 2013).

The sizes of the L- and R-ngrams are drawn independently at random. Each negative sample replaces either the L- or R-ngram of its corresponding positive (randomly, either *poh* or *jan* in Fig. 1) with characters drawn independently from the unigram distribution of the language of the sample.

All parameters, including embeddings, are initialized randomly. Dropout is applied to the embeddings and composed n-grams and a unit norm constraint is placed on the embeddings. We train using stochastic gradient descent with Adam learning rate adaptation, batch size 1000.

3.3 Validation criterion

The learned embeddings are affected by random initialization. As is typical in unsupervised learning, there is no simple way to select the best model, since we cannot evaluate the learned representations on a validation set. Conneau et al. (2017) define an *unsupervised validation criterion* to handle this problem in unsupervised alignment of word embeddings, which they use for model selection, as a proxy for word translation accuracy.

Eqn. 1 defines a validation criterion for a trained set of embeddings, *nn-sim*. To the extent that it correlates with the accuracy of correspondences found in the embeddings, it is suitable for model selection. To the extent that this holds throughout training, it can also be used for early stopping. We test these correlations in the next section. Given embeddings for languages A and B, we compute for each character in A the cosine similarity to its nearest neighbour from B, and take the mean over A's characters.

$$\textit{nn-sim} = \frac{1}{|A|} \sum_{a \in A} \min_{b \in B} cos(a, b) \qquad (1)$$

Eqn. 2 defines an evaluation metric *pair-rank* that can be computed where the desired pair correspondences $(a, b) \in C$ are known. It measures how well the correspondences are retrieved by the embeddings. For each (a, b), we compute the rank, by cosine distance from a, of b among all characters in B, normalized by the size of B. We compute the same in the opposite direction and take the average of all values. A lower value reflects a better retrieval of correspondences.

$$\textit{pair-rank} = \frac{1}{2|C|} \sum_{(a,b) \in C} \frac{rank_b(cos(a, B))}{|B|} + \frac{rank_a(cos(b, A))}{|A|} \qquad (2)$$

4 Experiments with artificial data

4.1 Motivation

For any pair of related languages, we expect to find a spectrum of correspondences between their characters, ranging from some very close pairs, through weaker correspondences, to no correspondence at all. There exists no gold-standard list of correspondences that a good model *should* find, making it difficult to evaluate representations.

(a) `Kaiken tämän lisäksi saan hellyyttä ja lämpöä sekä saan antaa sitä .`
(b) `Kêikei ëämän o?tänsi nêên hssGööëëä êê HämÞ?ä sÆiä sêën onë?ê siëä a`

Figure 2: Example sentence from the YLILAUTA corpus in its original form *(a)* and with the highest level of all three types of corruption *(b)*. The model is trained on an uncorrupted portion of the corpus as one language and a *distinct subset* to which this corruption has been applied.

We begin by testing XSYM on artificial datasets. We apply several types of corruption to real linguistic data, replacing some characters at random and combining or splitting others, then treat the corrupted data as a new language, with a distinct character set. The result is in some respects superficially similar to the relationship between related languages and presents similar challenges to the learning method. Crucially, having corrupted the data by known processes, we know which correspondences a successful method should recover.

First, we use corrupted data to measure how well the validation criterion *nn-sim* correlates with retrieval of known correspondences, measured by *pair-rank*. Then we analyze how robust the method is to the different types of corruption to get some insight into how it behaves.

4.2 Corruptions

We apply three different types of corruption. The input data has a character vocabulary V_i, the corrupted data V_o which may be different, since some corruptions add or remove characters. Corruptions are applied in the order presented. An example of the resulting text is given in Fig. 2.

Random noise: Randomly sample a given proportion p_{noise} of character tokens and, for each, sample a character at random to replace it with from the unigram distribution over V_i.

Systematic mapping: Systematically substitute a character a (randomly chosen from V_i) with b (randomly chosen from V_o). The resulting bs are indistinguishable from those that were bs in the input. a is now not in V_o, since it never occurs in the corrupted data. Characters are chosen for mapping until the expected proportion of tokens affected is $>p_{map}$. Since characters are sampled greedily to preserve randomness, the actual proportion, $\hat{p}_{map}$, may be greater than p_{map}.

Systematic splitting: Randomly choose a character a from V_o after the previous step, add new character b and randomly map half of as to b. Choose a number of characters in the same way as for mapping, until the expected proportion affected is $>p_{split}$. The actual proportion is $\hat{p}_{split}$.

Metric	PCC	Slope
p_{noise}	0.35	0.22
$\hat{p}_{map}$	0.67	0.36
$\hat{p}_{split}$	-0.12	-0.06
sum	0.52	0.11

Table 1: Pearson correlation coefficient and regression slope between the level of each type of corruption and the *pair-rank* evaluation metric.

We train embeddings using XSYM with two corpora, as if they represented different languages. The first is a randomly chosen subset of 95k documents from the YLILAUTA corpus of Finnish forum posts[2]. The second is a distinct subset of the same size, to which the corruptions have been applied. We run the training under different levels of each type of corruption, applying all 27 combinations of $p = 0, 0.15, 0.3$ for p_{noise}, p_{map} and p_{split}.

In the first experiment, we measure the correlation between *nn-sim* and *pair-rank*. We train each model once for exactly 10 corpus iterations, outputting both metrics every 500k samples, resulting in 70 measures per model. In the second, we train all models again, using *nn-sim* as an unsupervised criterion for early stopping and model selection over 5 random intializations.

4.3 Results

Testing validation criterion *nn-sim*. We find a Pearson correlation coefficient (PCC) of $r = 0.79$ between *nn-sim* and *pair-rank* from the 1,890 measurements taken during training. The high correlation suggests that *nn-sim* is a good criterion to use for early stopping. Furthermore, measuring only at the end of training, we get $r = 0.83$, supporting the use of *nn-sim* to choose between embeddings from alternative initializations. We can expect that embeddings that maximize *nn-sim*

[2] http://urn.fi/urn:nbn:fi:lb-2015031802

would also have maximized (or close) *pair-rank*, had we been able to measure it using known correspondences.

Testing effect of corruptions. Training all models with early stopping and model selection, we measured the correlation between the level of each corruption (and the sum of the three) and the *pair-rank* of the final embeddings (Table 1). We also report the slope of the regression between the corruption levels and *pair-rank*. Values of *pair-rank* range from 6%, for a low level of corruption, to 37% for a high level, with a mean of 16% over all 27 tests.

There is a high correlation for character mapping: the more characters are conflated with others in the vocabulary, the harder it is to identify the correspondences. This is unsurprising: to maintain the same level of accuracy after a mapping $a \Rightarrow b$, the method must recognize the similarity in the contexts of *2:b* in the corrupted data to those of both *1:a* and *1:b* in the uncorrupted data. The contextual distribution of *2:b*'s usage is in effect the average of those of *1:a* and *1:b*, so becomes hard to identify with either.

There is a relatively low correlation for random noise. The method is robust to this corruption, which obscures the regularities in the data, but has no systematic effect on the contextual distributions of any of the symbols.

There is no correlation for character splitting. When *1:a* is split at random so that it appears as either *2:a* or the newly added *2:b*, both *2:a* and *2:b* can be expected to have similar contextual distributions to *1:a*. The splitting reduces the amount of data from which to infer the distributions, but does not prevent the model from discovering the similarity, even under high levels of other corruptions.

These results suggest promisingly that XSYM is effective at recovering correspondences between symbols in two datasets where there are similarities in the symbols' contexts of use. It is impossible to know how these different types and levels of corruption correspond to the difficulties the method faces dealing with real data. However, this experiment confirms that the model is discovering and exploiting the sort of distributional similarities that we would hope, even where the contextual distributions are not directly comparable.

5 Experiments with linguistic corpora

We now apply XSYM to real linguistic data. To ensure that the method is not exploiting similarities between two corpora due to a shared domain (e.g., prevalence of particular cognate words peculiar to that domain), we apply it to corpora from unrelated domains, as well as in-domain pairs.

We first compare Finnish and Estonian. Whilst not low-resourced languages, it is easier to interpret results from these well-studied, closely related languages, and they are a good starting point for studying low-resourced Uralic languages. For Finnish, we use the YLILAUTA corpus again. For Estonian, we use the newspaper portion of the Estonian Reference Corpus, balanced subcorpus (Kaalep et al., 2010, henceforth EST-REF-NEWS). We use only the first 190k documents in Ylilauta, to match the size of EST-REF-NEWS (∼5.8M tokens). We lower-case the text to simplify analysis and treat very rare characters ($<$ 500 occurrences) as a single out-of-vocabulary token. We also run on a single-domain corpus pair, to see how the outcome is affected by comparable versus non-comparable corpora. We train on YLILAUTA together with the forum portion of the Estonian Reference Corpus (∼6.4M tokens, henceforth EST-REF-FORUM). Training parameters are identical to the previous section and *nn-sim* is used for early stopping and model selection.

We also apply the method to several combinations of low-resourced Uralic (North Finnic) languages: two dialects of Karelian (Olonets and North Karelian) and the severely endangered Ingrian language (∼130 speakers). All corpora are Bible translations from the University of Helsinki Corpus Server[3], with ∼150k, 200k and 30k tokens respectively. We report metrics for some pairs within low-resourced languages and also for Ingrian–Finnish, since many applications will involve comparing a low-resourced language to a better-resourced one.

5.1 Results

Since this is an unsupervised learning task and there is no gold-standard set of correspondences, we cannot directly evaluate the embeddings quantitatively. Ultimately, their value will be tested by their usefulness in a downstream task, such as cognate discovery, but we leave this to future work.

[3] http://urn.fi/urn:nbn:fi:lb-201403269

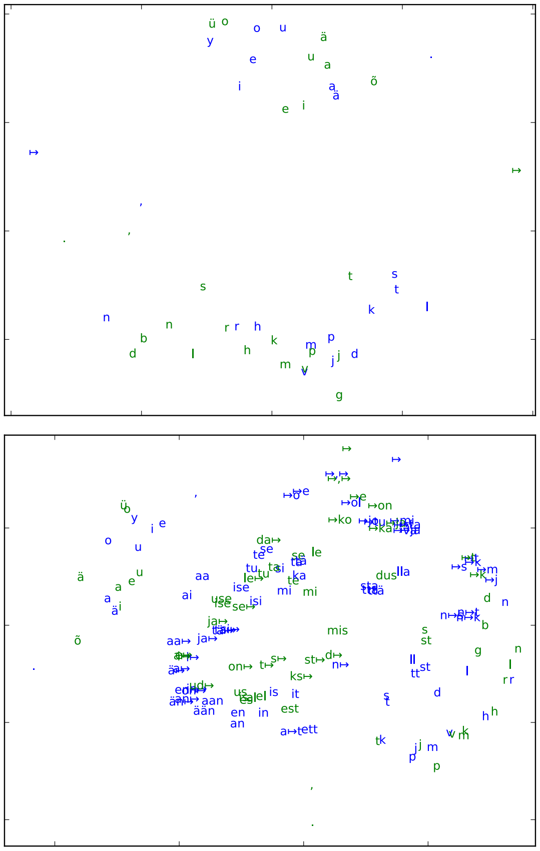

Figure 3: MDS reductions of mixed-domain embeddings for Finnish (blue) and Estonian (green). Plot of individual characters (top) and most frequent character bigrams and trigrams (bottom). '↦' represents space.

Fig. 3 shows reductions to 2D using multidimensional scaling (MDS) of the embeddings trained on Finnish and Estonian with mixed domains. We show a plot of the embeddings for all individual characters and another including the most frequent character bigrams and trigrams in each language. Fig. 4 shows single-character embeddings for single-domain corpora.

The plots give a broad notion of the layout of the space, but poorly reflect proximity between individual pairs. We also present statistics about the proximity of common characters with frequency $\geq 0.5\%$ in both corpora (e.g. *fi:t–et:t*) in Table 2. We measure where *et:t* appears in a ranking of all Estonian characters by proximity to *fi:t*, and average over all pairs, in both directions. We also report the percentage of cases where the identical character is the nearest (R@1) and within the nearest 3 characters (R@3) in the other language.

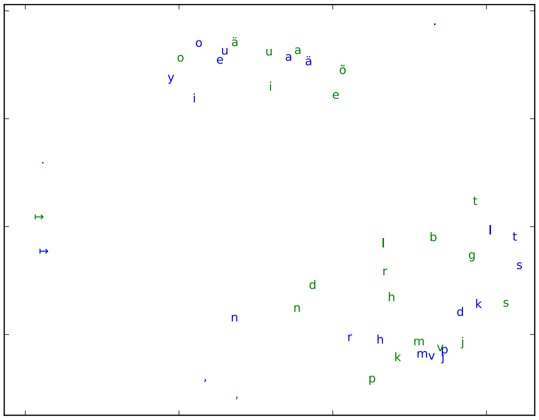

Figure 4: MDS reduction of single-domain, forum post embeddings for Finnish (b) and Estonian (g).

Importantly, this is *not an evaluation metric*, but rather a sanity check: a lower value does not necessarily reflect better embeddings, since there may be good reasons to map non-identical characters close to each other. (Indeed, this is one of the motivations for our approach.) However, the fact that the ranking is typically low is an encouraging sign that the method is succeeding in discovering meaningful correspondences between the languages. Moreover, we see no clear difference in this respect between cross-domain and in-domain learning. The results for Uralic languages demonstrate the applicability of the method to small datasets for low-resourced languages.

To give further insight into what is being captured, in Table 3 we present, for two language pairs, nearest neighbours across languages for all cases where the nearest was not the identical character. Of particular interest here are the discovered close correspondences *š–s* and *y–ü* between North Karelian and Olonets. Table 4 shows, for one pair in one direction, other near neighbours where the nearest *is* the same character.

6 Future work

We plan to perform extrinsic evaluation of learned embeddings, like Tsvetkov et al. (2016), testing the embeddings on downstream tasks. One example is cognate discovery, where the learned similarities may bring advantages over the assumed or initial correspondences used in related work, for example where distinct symbol sets are used. Learned similarities can be incorporated into many existing cognate discovery methods (e.g. Kondrak, 2009), with McCoy and Frank (2018)

Corpus 1	**Corpus 2**	**Chars 1**	**Chars 2**	**Common**	**MPR**	**R@1**	**R@3**
Ylilauta	Est-Ref-News	23	26	22	2.32	55%	82%
Ylilauta	Est-Ref-Forum	23	25	25	2.11	57%	84%
North Karelian	Olonets Karelian	24	26	22	1.20	89%	95%
Ingrian	Olonets Karelian	22	26	21	1.74	64%	90%
Ingrian	Ylilauta	22	23	22	1.73	80%	89%

Table 2: Correspondence between common characters for cross-domain and in-domain models, as a sanity check. **Chars 1** and **2** are the number of characters in each language's vocabulary after the frequency filter. **Mean pair rank (MPR)**: mean rank of a character by cosine similarity to its identical character in the other language. **R@1** is the proportion that are nearest neighbours, **R@3** the proportion that are within the three closest.

Ing	*Fi*
h	v h
r	h v m r
,	n ,
.	n , s i ä .

Fi	*Ing*
d	k j v ...
t	k t
.	, o s .

NK	*Olonets*
š	k s p ...
s	l z g s
y	ü ä e ...
,	. ,

Olonets	*NK*
d	j t l ...
ä	e ä
g	l j s ...
s	š k p s
z	l j r ...
ü	y ä e ...
,	. ,

Table 3: Nearest neighbours across Finnish (Ylilauta)–Ingrian and North Karelian–Olonets, where the closest is not the same. **Bold** are not in the other language.

NK	Olonets							
a	a	o	u	ü	e			
v	v	j	p	m	r	k	t	
ä	ä	ö	e	ü	a			
e	e	ä	o	a	ü			
i	i	ü						
h	h	n						
k	k	s	p	m	v	j	t	
j	j	v	m	p	d	k	r	z
m	m	v	r	p	j			
l	l	z	g	n	r			
o	o	a	u	e				
n	n	h	l					
p	p	v	m	r	j	k		
r	r	v	m	p	h	j		
u	u	o	a					
t	t	j	d	k				
ö	ö	ä						

Table 4: Nearest Olonets neighbours to North Karelian, where the nearest is identical, down to a cosine similarity of 0.5.

providing perhaps a particularly suitable way to use them. It remains an open question how n-gram similarities can be used here. Another possible application is spelling translation, for example applying the method of Pirkola et al. (2003) without requiring translation dictionaries.

A potential benefit of this method is its ability to capture correspondences between different lengths of n-grams, not just individual symbols. In our analysis (Fig. 3) we have used this by including a language's most common n-grams in projections, but other ways to select pertinent correspondences are possible, for example taking into account similarities or the structure of the vector space as well as frequency.

Xsym is similar to Polyglot language models (Tsvetkov et al., 2016). We have suggested, but not demonstrated here, that it is better suited to direct comparison of symbols. Investigation of the properties of representations learned by the two methods is required and we will test Xsym on the tasks reported by Tsvetkov et al. (2016).

We plan to apply Xsym to other symbol sequences, in particular, to sequences of phonetic symbols from speech (like List, 2014). It may be possible to use automatic transcriptions that do not require language-specific transcribers, since the symbols need not correspond to linguistically motivated systems, such as IPA. Although designed for learning about linguistic sequences, Xsym could potentially also be applied also to non-linguistic data to discover links between sequences that use distinct vocabularies. We will investigate what characteristics of sequences are essential in finding useful abstractions (e.g. vocabulary size).

7 Conclusion

We have presented an unsupervised method that uses a neural network to learn vector representations of symbols and short n-grams on the basis of their contexts observed in sequences. It is able to learn comparable representations of symbols from multiple languages that use distinct symbol sets, learning to exploit similarities in the context distributions of the symbols across languages, even though the symbols in the contexts are also drawn from distinct vocabularies.

We have demonstrated the method's ability to recover mappings between vocabularies, even when they are obscured by ambiguity in the mappings and noise, provided that the noise does not obscure the distributions over the symbols' contexts too much. We then showed some results of applying the method to real linguistic data, focusing here on characters in text and several Uralic language pairs. We found that it was able to recognize many characters that are common to the corpus pairs as being closely related by their contexts of use. An even closer correspondence was found between closely related, low-resourced dialects, despite a much smaller training set.

The learned similarities between symbols provide a way to bootstrap discovery of other linguistic similarities, such as morphology or cognate words. We leave testing on these applications to future work and have presented here some analysis of the learned representations, which appear highly promising. We suggest that the results have great potential as a first step in fully unsupervised linguistic typology. Discovered correspondences may also be able to tell us about typology in themselves. For example, some measures of orthographic difference and sound correspondences correlate with geographic factors in language development (Heeringa et al., 2013; Prokić and Cysouw, 2013). Discovered strong symbol correspondences (especially if the method is applied to phonetic sequences) could also be of typological interest in themselves.

The method presented is a generic representation learning technique for symbol sequences. As well as text, it could also be applied to other linguistic sequences, such a phonetic transcriptions, and potentially even to non-linguistic sequences. On the basis of the encouraging initial results presented here, we suggest that it warrants further investigation, including linguistic applications, such as unsupervised cognate discovery, and other aspects of linguistic typology.

8 Acknowledgements

This work was funded by the Academy of Finland *Digital Language Typology* project (no. 12933481).

References

Vladimir Batagelj, Tomaž Pisanski, and Damijana Keržič. 1992. Automatic clustering of languages. *Computational Linguistics*, 18(3):339–352.

Chris Brew, David McKelvie, et al. 1996. Word-pair extraction for lexicography. In *Proceedings of the 2nd International Conference on New Methods in Language Processing*, pages 45–55.

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. 2012. Turning the pipeline into a loop: Iterated unsupervised dependency parsing and PoS induction. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, pages 96–99.

Alina Maria Ciobanu and Liviu P Dinu. 2014. Automatic detection of cognates using orthographic alignment. In *Proceedings of the 52nd Annual Meeting of the ACL*, volume 2, pages 99–105.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *CoRR*, abs/1710.04087.

Antonella Delmestri and Nello Cristianini. 2010. String similarity measures and PAM-like matrices for cognate identification. *Bucharest Working Papers in Linguistics*.

Grzegorz Kondrak Diana Inkpen, Oana Frunza. 2005. Automatic identification of cognates and false friends in French and English. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 251–257.

Bergsma Fenna, Swarte Femke, and Gooskens Charlotte. 2014. Does instruction about phonological correspondences contribute to the intelligibility of a related language? *Dutch Journal of Applied Linguistics*, 3(1):45–61.

Pablo Gamallo, José Ramom Pichel, and Iñaki Alegria. 2017. From language identification to language distance. *Physica A: Statistical Mechanics and its Applications*, 484:152–162.

Luís Gomes and José Gabriel Pereira Lopes. 2011. Measuring spelling similarity for cognate identification. In *Proceedings of the Portuguese Conference on Artificial Intelligence*, pages 624–633.

David Hall and Dan Klein. 2010. Finding cognate groups using phylogenies. In *Proceedings of the 48th Annual Meeting of the ACL*, pages 1030–1039.

Wilbert Heeringa, Jelena Golubovic, Charlotte Gooskens, Anja Schüppert, Femke Swarte, and Stefanie Voigt. 2013. Lexical and orthographic distances between Germanic, Romance and Slavic languages and their relationship to geographic distance. *Phonetics in Europe: Perception and Production*, pages 99–137.

Heiki-Jaan Kaalep, Kadri Muischnek, Kristel Uiboaed, and Kaarel Veskis. 2010. The Estonian Reference Corpus: Its composition and morphology-aware user interface. In *Proceedings of the 4th International Conference Baltic HLT*, pages 143–146.

Kenji Kita. 1999. Automatic clustering of languages based on probabilistic models. *Journal of Quantitative Linguistics*, 6(2):167–171.

Grzegorz Kondrak. 2000. A new algorithm for the alignment of phonetic sequences. In *Proceedings of the 1st NAACL*, pages 288–295.

Grzegorz Kondrak. 2009. Identification of cognates and recurrent sound correspondences in word lists. *TAL*, 50:201–235.

Grzegorz Kondrak and Graeme Hirst. 2002. *Algorithms for language reconstruction*. Ph.D. thesis, University of Toronto.

Jonas Kuhn. 2004. Experiments in parallel-text based grammar induction. In *Proceedings of the 42nd Annual Meeting of the ACL*.

Johann-Mattis List. 2012. Lexstat: Automatic detection of cognates in multilingual wordlists. In *Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH*, pages 117–125.

Johann-Mattis List. 2014. *Sequence comparison in historical linguistics*. Ph.D. thesis, Heinrich-Heine-Universität Düsseldorf. Dissertations in Language and Cognition, 1.

Richard T. McCoy and Robert Frank. 2018. Phonologically informed edit distance algorithms for word alignment with low-resource languages. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2018*, pages 102–112.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Andrea Mulloni. 2007. Automatic prediction of cognate orthography using support vector machines. In *Proceedings of the 45th Annual Meeting of the ACL: Student Research Workshop*, pages 25–30.

Andrea Mulloni and Viktor Pekar. 2006. Automatic detection of orthographic cues for cognate recognition. *Proceedings of LREC'06*.

Mirabela Navlea and Amalia Todirascu. 2011. Using cognates in a French-Romanian lexical alignment system: A comparative study. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 247–253.

Robert Östling and Jörg Tiedemann. 2016. Continuous multilinguality with language vectors. *CoRR*, abs/1612.07486.

Filippo Petroni and Maurizio Serva. 2008. Language distance and tree reconstruction. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(08):P08012.

Ari Pirkola, Jarmo Toivonen, Heikki Keskustalo, Kari Visala, and Kalervo Järvelin. 2003. Fuzzy translation of cross-lingual spelling variants. In *Proceedings of the 26th Annual International ACM SIGIR Conference*, pages 345–352.

Jelena Prokić and Michael Cysouw. 2013. Combining regular sound correspondences and geographic spread. *Language Dynamics and Change*, 3(2):147–168.

Jelena Prokić, Martijn Wieling, and John Nerbonne. 2009. Multiple sequence alignments in linguistics. In *Proceedings of the EACL 2009 Workshop on Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education*, LaTeCH-SHELT&R '09, pages 18–25.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of NAACL HLT 2013*, pages 74–84.

Benjamin Snyder, Tahira Naseem, and Regina Barzilay. 2009. Unsupervised multilingual grammar induction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 73–81.

Yulia Tsvetkov, Sunayana Sitaram, Manaal Faruqui, Guillaume Lample, Patrick Littell, David R. Mortensen, Alan W. Black, Lori S. Levin, and Chris Dyer. 2016. Polyglot neural language models: A case study in cross-lingual phonetic representation learning. *CoRR*, abs/1605.03832.