Cross-Family Similarity Learning for Cognate Identification in Low-Resource Languages

Soisalon-Soininen, Eliel

2019-09-04

Soisalon-Soininen, E & Granroth-Wilding, M 2019, Cross-Family Similarity Learning for Cognate Identification in Low-Resource Languages. in G Angelova, R Mitkov, I Nikolova & I Temnikova (eds), RANLP 2019 - Natural Language Processing a Deep Learning World : Proceedings . INCOMA, Shoumen, pp. 1121-1130, Recent Advances in Natural Language Processing, Varna, Bulgaria, 02/09/2019.

http://hdl.handle.net/10138/307967

Cross-Family Similarity Learning for Cognate Identification in Low-Resource Languages

Eliel Soisalon-Soininen
Department of Computer Science
University of Helsinki
eliel.soisalon-soininen@helsinki.fi

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

Abstract

We address the problem of cognate identification across vocabularies of any pair of languages. In particular, we focus on the case where the examined languages are low-resource, to the extent that no training data whatsoever in these languages, or even closely related ones, is available for the task. We investigate the extent to which training data from another, unrelated language family can be used instead. Our approach consists of learning a similarity metric from example cognates in Indo-European languages and applying it to low-resource Sami languages of the Uralic family. We apply two models, following previous work: a Siamese convolutional neural network (S-CNN) and a support vector machine (SVM), and compare them with a Levenshtein distance baseline. We test performance on three Sami languages and find that the S-CNN outperforms the other approaches, suggesting that it is better able to learn such general characteristics of cognateness that carry over across language families. We also experiment with fine-tuning the S-CNN model with data from within the language family in order to quantify how well this model can make use of a small amount of target-domain data to adapt.

1 Introduction

Cognate identification is a core task in the *comparative method*, a collection of techniques used in historical linguistics for the inference of language family trees, reconstruction of protolanguages, and other areas of study related to language history (List, 2013). Cognate information can also be used to improve natural language processing (NLP) applications, such as machine translation (Grönroos et al., 2018). In addition, knowledge of cognates can be useful for second-language learning (Beinborn et al., 2014).

For a subset of the world's languages, such as Indo-European, language-family trees, protolanguages, and etymological databases are well-established. However, the majority of languages have only few speakers, and such resources are scarce. Since the tasks of the comparative method are laborious to do manually, computational approaches have been taken to automatize these tasks. In addition to cognate identification, previous work addresses phonetic alignment (Kondrak, 2000; Prokić et al., 2009; List, 2013), inference of family trees (Chang et al., 2015; Jäger, 2014; Bouckaert et al., 2012), and reconstruction of proto-words (Bouchard-Côté et al., 2013).

Ideally, computational approaches to historical linguistics should be applicable to any language, even in the absence of hand-crafted resources and analyses. Recent work addressing cognate identification for *low-resource* languages assumes either the existence of high-resource relatives, to be used as training data (McCoy and Frank, 2018), or the availability of detailed dictionary definitions (St Arnaud et al., 2017).

In this paper, we address cognate identification in a scenario where we are only given a set of unannotated vocabularies from truly low-resource languages, namely South, North, and Skolt Sami of the Uralic family, without the aforementioned resources. We only assume a training dataset of example cognates in Indo-European languages, highly unrelated to our languages of interest. It might be expected that knowledge of general tendencies in patterns of correspondence between related languages, such as common phoneme substitutions, might be of some use, even

Word *x*	Word *y*	Meaning of *x*	Meaning of *y*
it: *notte*	es: *noche*	'night'	'night'
en: *attend*	fr: *attendre*	'attend'	'wait'
fi: *huvittava*	et: *huvitav*	'amusing'	'interesting'
en: *oath*	sv: *ed*	'oath'	'oath'
fi: *pöytä*	sv: *bord*	'table'	'table'
en: *bite*	fr: *fendre*	'bite'	'split'

Table 1: Examples of cognates, i.e. etymologically related words. The degree of similarity in form and meaning may vary quite substantially.

when searching for potential cognates in a different language family. Naturally, some knowledge of more closely related languages, or of the language pair in question, is more informative, and we attempt to quantify how well one of these models is able to make use of that.

Our aim is to investigate the extent to which a similarity learning approach, that is learning a similarity metric in a data-driven manner, is able to generalize across language families. We experiment with two similarity learning approaches from previous work, namely a support vector machine (SVM, Hauer and Kondrak, 2011) and a Siamese convolutional neural network (S-CNN, Rama, 2016), compared with a Levenshtein distance baseline (LD, Levenshtein, 1966). We train the models on examples of cognates in Indo-European language pairs, then test how well they are able to identify cognates in the Sami language pairs, not seen at training time. In addition, we fine-tune the S-CNN model on labelled target-language pairs, in order to quantify how much the lack of target-family training data affects performance.

Next, we explain the cognate identification problem and its difficulties, and review previous approaches to the problem. Then we present the approaches we use in our experiments, as well as the experimental setup in more detail. Finally, we analyse the results of the experiments.

2 The cognate identification problem

The term *cognate* has several distinct uses in the literature. In historical linguistics, two words are considered cognates only if they have descended from the same ancestor word in a common proto-language, implying that they also belong to two related languages (e.g. Jäger et al., 2017; List, 2013; Kondrak, 2009). Meanwhile, a number of broader definitions have been used in NLP, motivated by practical concerns. For example, some authors refer to any etymologically related pair of words (i.e. sharing a common origin) as cognates, including, for example, loanwords (e.g. Kondrak, 2001; Beinborn et al., 2013; Bloodgood and Strauss, 2017). Others assume that cognates share both a similar form and common meaning (e.g. Nakov and Tiedemann, 2012; Bergsma and Kondrak, 2007). This assumption is problematic for historical linguistics, since it excludes cognate words that have come to have different meanings since the languages diverged, but it may be more useful for some language learning applications. In this paper, we regard any pair of etymologically related words as cognates, including genetically related true cognates as well as direct loanwords or loans from a common origin.

We formulate the cognate identification problem as follows. We are given two string sets $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$. The task is to extract those pairs (x, y) in relation R:

$$R = \{(x, y) \in X \times Y |\ x \text{ is cognate with } y\ \}.$$

Each element $x \in X$ and $y \in Y$ is a string over alphabets Σ_x and Σ_y respectively. The alphabets do not necessarily overlap, since the orthographies of different languages may vary. This issue is often circumvented by using phonetic transcriptions of words. Lacking phonetic transcriptions for our test data, we deal with orthographic forms. Nonetheless, orthographic similarity often reflects phonetic similarity, in particular in the Sami languages we take as examples here, where the orthography is largely phonemic.

Several factors have been found to predict cognateness: phonetic similarity (reflected by orthographic similarity), semantic similarity, and the presence of *regular sound correspondences*, word segments regularly occurring in similar phonetic positions and contexts (Kondrak, 2009).

The example cognates in Table 1 illustrate the difficulty of cognate identification. A straightforward example is the Italian–Spanish pair *(notte, noche)*, with a similar form and common meaning. However, many cognates have similar surface forms, but differ in meaning, such as the English–French *(actual, actuel)* and Finnish–Estonian *(huvittava, huvitav)*. Such words are referred to as *false friends* in the context of language learning.

Furthermore, cognates might look very different on the surface. English–Swedish cognates *(oath, ed)* and Finnish–Swedish *(pöytä, bord)* look quite

different, but share a meaning (and common origin). On the other hand, English–French *(bite, fendre)* are similar neither in form nor meaning. The only way to recognise such cognates from their surface forms alone is to identify regular correspondences, such as *th – d* for English–Swedish.

Consequently, and in contrast to much previous work, we make no strict assumptions about the degree of similarity in form or meaning that any two cognates should exhibit. Instead, following Jäger (2014), we treat regular correspondences as the main driving factor in the cognate relation and attempt to capture these in a completely data-driven manner.

3 Related work

Earlier computational approaches to cognate identification attempt to design a string similarity (or distance) metric that assigns a higher score to cognate words and a lower score to unrelated ones. A common approach is to extend the traditional Levenshtein distance (Levenshtein, 1966) by associating specific weights to pairs of symbols using linguistic knowledge (e.g. Kondrak, 2000; List, 2013), or sets of example cognates (e.g. Bergsma and Kondrak, 2007; Rama, 2015).

Kondrak (2000) proposes the ALINE algorithm using specific weights based on several predetermined phonetic features. In addition, Kondrak (2005) generalizes the Levenshtein distance with the n-gram similarity measure. Turchin et al. (2010) use a heuristic based on mapping consonants to ten classes, and consider words matching in their first two consonant classes to be cognates. The SCA algorithm of List (2013) uses a larger set of sound classes and also considers prosodic aspects of words.

Other authors rely on learning regular correspondences (sometimes called *mismatches* or *substitution patterns*) from example cognates using an alignment algorithm. For example, Ciobanu and Dinu (2014) and Bergsma and Kondrak (2007) use a global alignment algorithm to align orthographic word pairs and extract substring pairs, which they use as features for an SVM. Gomes and Pereira Lopes (2011) use the same approach to develop a weighted string similarity metric for words in orthographic form. Rama (2015) use gap-weighted subsequences as features. McCoy and Frank (2018) use character embeddings and cosine similarity to extend Levenshtein distance.

Hauer and Kondrak (2011) convert word pairs into features for an SVM using a set of string similarity metrics. This approach has been extended with features for semantic similarity, for example using the lexical database WordNet (Jäger et al., 2017; St Arnaud et al., 2017; Kondrak, 2009). Bloodgood and Strauss (2017) improve further such an SVM model using global constraints and reranking. In addition, St Arnaud et al. (2017) utilise English and Spanish word embeddings of dictionary definitions. This SVM classification approach is one of the methods that we apply to cross-language family learning.

Jäger (2014) and Rama (2016) take data-driven approaches not relying on hand-designed features. Jäger proposes a similarity metric based on weights for symbol pairs given by *pointwise mutual information*, the values for which were learned from a training set of cognate pairs. Rama applies deep learning, encoding words into a grid-like representation and applying a Siamese convolutional neural network to cognate identification for multilingual wordlists. He uses two methods to encode a phonetic symbol into vector, a one-hot encoding and one based on phonetic features, achieving better performance with one-hot encodings for two out of three language families. This approach is another method in our comparison of models for cross-language family learning.

In recent work, Hämäläinen and Rueter (2019) take an alternative approach of applying neural machine translation methods to the problem of predicting a cognate given a word in a related language. The same model could in principle be applied to the task we present here and we intend to make a direct comparison in future work.

4 Methods

In this section, we present the three approaches to solving the cognate identification problem that we have used in our experiments: a string similarity metric based on the Levenshtein distance (Levenshtein, 1966) used as a baseline, an SVM with several string similarity metrics as features (Hauer and Kondrak, 2011), and a Siamese convolutional neural network (Rama, 2016).

4.1 Levenshtein distance–based similarity

The *Levenshtein distance* $d_L(s_1, s_2)$, or *string edit distance*, between strings s_1 and s_2 over an alphabet Σ is the minimum number of insertion, dele-

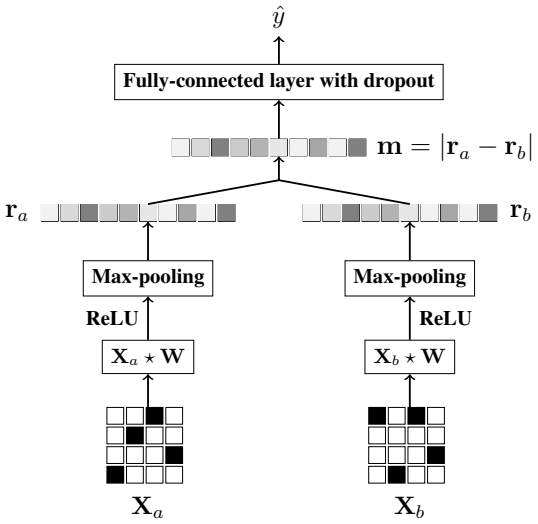

Figure 1: Architecture of the S-CNN. Column vectors in input matrices represent one-hot-encoded characters. The same filter $\mathbf{W}$ is convolved with both inputs.

tion, or substitution operations needed to transform one string to the other. To obtain the *normalised* Levenshtein distance, this number is divided by the length of the longer word, equal to the maximum possible distance between s_1 and s_2. The similarity metric is then:

$$sim_L = 1 - \frac{d_L(s_1, s_2)}{max(|s_1|, |s_2|)}.$$

For example, for the cognate pairs *(coupe, Kopf)* and *(pöytä, bord)*, the respective similarities are $1 - \frac{3}{5} = 0.4$ and $1 - \frac{5}{5} = 0$. It is assumed that both strings are drawn from overlapping alphabets, since the similarity is always zero for disjoint alphabet sets.

Previous work has introduced a variety of Levenshtein-based measures by defining different ways of learning or computing the cost associated with a character substitution. Here we apply the basic version, in which a matching pair of characters have a zero cost and any other a unit cost.

4.2 Support vector machine

The support vector machine (SVM) is a supervised learning model trained by finding the optimal separating hyperplane between multi-dimensional data points of different classes. The basic SVM is a non-probabilistic, linear binary classifier. For data that is linearly separable, the optimal hyperplane creates the maximum margin between training points in the two classes. When the data classes are not linearly separable, the margin can still be maximised while allowing some data points to be on the wrong side of the optimal hyperplane. Another approach is to use a non-linear kernel function, which enlarges the feature space using basis expansions, such as a polynomial or a radial-basis function.

For the model comparison in our experiments, we have implemented the SVM model used by Hauer and Kondrak (2011). In this model, a pair of strings (s_1, s_2) is represented by a feature vector $\mathbf{x} \in \mathbb{R}^6$ such that

- x_1 is the Levenshtein distance $d_L(s_1, s_2)$,
- x_2 is the number of common bigrams,
- x_3 is the prefix length,
- x_4 is the length of s_1,
- x_5 is the length of s_2, and
- x_6 is the absolute difference between the lengths, i.e. $x_6 = |x_4 - x_5|$.

We have chosen this SVM model as it is based on string similarity measures that are applicable to the low-resource language setting. More recent SVM-based approaches to cognate identification exist, but they either require detailed dictionary definitions in a high-resource language with high-quality pre-trained word embeddings (St Arnaud et al., 2017), or multilingual word lists aligned by concepts (Jäger et al., 2017).

4.3 Siamese convolutional neural network

The Siamese convolutional neural network (S-CNN) is a supervised learning model originally proposed by Chopra et al. (2005) for the task of face verification, and applied with some modification to cognate identification by Rama (2016). Our implementation is based on the latter model. The architecture is presented in Figure 1.

The S-CNN is a two-input version of the convolutional neural network (CNN) specialized in processing data with a grid-like topology. CNNs have been very successful in computer vision, and they have also been applied to several NLP tasks, such as text classification (e.g. Zhang et al., 2015).

When applied to NLP tasks, the CNN requires a grid-like representation of the input. In the case of cognate identification, it is convenient to represent a word as a matrix $\mathbf{X} \in \{0, 1\}^{|\Sigma| \times n}$ such

Dataset	# cognate	# all pairs	$\lvert\Sigma\rvert$
IE-TRAIN	73,238	732,380	329
sma–sme	1,460	11,234 × 47,312	42 (27)
sma–sms	838	11,234 × 29,401	75 (27)
sme–sms	2,188	47,312 × 29,401	77 (38)

Table 2: The datasets used in the experiments. Etymological WordNet is used for training, and other datasets are used for testing (see Table 3 for smaller fine-tuning sets). $|\Sigma|$ is the number of all characters observed in a dataset. The number of overlapping characters is given in parentheses (for language pairs). Languages: South Sami (sma), North Sami (sme), Skolt Sami (sms).

Dataset	# cognate	# all pairs
SAMI-FT	986	100,000
SAMI-FT-TEST	3,500	350,000

Table 3: The small-scale datasets sampled from the Sami vocabularies in Table 2. We use these in experiment 2 to fine-tune the S-CNN and analyse how the number of in-family training pairs affects the performance.

that $\mathbf{X} = [\mathbf{x}_1\mathbf{x}_2 \ldots \mathbf{x}_n]$, where each column vector $\mathbf{x}_i \in \{0,1\}^{|\Sigma|}$ is a one-hot vector representing a character in the alphabet Σ. The training data $D = \{(\mathbf{X}_{ai}, \mathbf{X}_{bi}), y_i\}_{i=1}^{N}$ then consists of pairs of words such that $y_i = 1$ if $\mathbf{X}_{ai}$ and $\mathbf{X}_{bi}$ are cognates, and $y_i = 0$ otherwise.

As shown in Figure 1, the S-CNN model is an extension of the CNN: first, one filter $\mathbf{W} \in \mathbb{R}^{p \times q}$ is convolved (cross-correlated) over character sequences of length q from both input matrices $\mathbf{X}_a$ and $\mathbf{X}_b$, producing a feature map for each input matrix. These are run through a rectified linear function, whereafter max-pooling is applied to the results. The number of rectified and max-pooled feature maps produced from each input matrix is equal to the number of filters. We fix the filter height at $p = |\Sigma|$, equal to the size of the alphabet and the height of the input matrix.

The representation vectors $\mathbf{r}_a$ and $\mathbf{r}_b$ are obtained by concatenating all the feature maps into single vectors. These vectors are then merged into one vector $\mathbf{m}$ using some distance metric. We use the absolute vector difference such that $\mathbf{m} = |\mathbf{r}_a - \mathbf{r}_b| = [|r_{a1} - r_{b1}|, |r_{a2} - r_{b2}|, \ldots, |r_{al} - r_{bl}|]^T$, where $l = |\mathbf{r}_a| = |\mathbf{r}_b|$. Finally, the merged vector $\mathbf{m}$ is fed as input to a fully-connected layer, itself connected to the output neuron. The dropout technique of Srivastava et al. (2014) is applied to the fully-connected layer, and the output neuron is activated with the sigmoid function. The output of a trained model can be regarded as a learned similarity metric between pairs of inputs.

5 Experiments

In this section, we present our datasets, experimental setup, training and fine-tuning procedures, and our evaluation scheme.

5.1 Datasets

A summary of the datasets is shown in Table 2. All source data for training and testing is publicly available and we release the exact processed training and test sets for reproducibility[1].

We use the Etymological WordNet (Gerard de Melo, 2014) as our training data for the SVM and S-CNN models. This is a database containing information of etymological origin, cognateness, as well as derivational and compositional links between words. The database consists of word pairs that each belong to one of the aforementioned relations. The database has been mined from Wiktionary, and its entries are mostly from widely spoken Indo-European languages.

Since we are concerned with the identification of cognates across languages, we only use as our training data those word pairs that are either cognates, or where one word is the root of the other. Thus, we exclude derivationally and compositionally linked word pairs from our training set. Furthermore, we filtered out those pairs where both words belong to the same language. In total, there were 73,238 cognate pairs in the filtered training set. In order to train a discriminative classifier, we generated negative examples by randomly pairing unrelated words, so that the ratio of cognate to unrelated word pairs was 10%. We refer to the resulting training set as IE-TRAIN.

[1] All datasets released at https://github.com/soisalon/LRCognates.

As a source of unannotated word lists from low-resource languages, we use a set of three vocabularies from North, South, and Skolt Sami of the Uralic family. We have retrieved these vocabularies from dictionaries compiled by Giellatekno[2]. We filtered out all words with upper-case (proper nouns) or non-alphabetic characters. We retrieved gold-standard cognate sets for evaluation and fine-tuning from Álgu[3], the etymological database for Sami languages. This database contains (positive-only) cognate information for only a subset of all the words in the vocabularies. We refer to this dataset as SAMI-FULL and average results over the three pairs of languages. The evaluation scheme is explained in detail in section 5.3.

In addition, to fine-tune and evaluate models in experiment 2 (see section 5.4), we sample small-scale sets with a higher proportion (1%) of cognates, presented in Table 3, SAMI-FT and SAMI-FT-TEST.

5.2 Training and fine-tuning

In our implementation of the S-CNN model, we used ten filters with width $q = 2$ and height $p = |\Sigma|$ (alphabet size). The alphabet was the set of all characters observed in both the training and test datasets, and its size was $|\Sigma| = 336$. We fixed the input matrix width to $n = 20$. For words shorter than this, the input matrices were zero-padded, and longer words were truncated at this length. In the fully-connected layer, we used a dropout rate of 0.5.

We trained the S-CNN model using binary cross-entropy as the loss function, and the Adadelta optimizer (Zeiler, 2012) with initial learning rate $\alpha = 1.0$, decay rate $\rho = 0.95$, and the constant $\epsilon = 1 \cdot 10^{-6}$. The batch size was set at 128, and number of epochs was 50. In fine-tuning (experiment 2), the respective values were 32 and 20. Otherwise, we used the same hyperparameters when fine-tuning the model. We implemented the model using the Keras library with Tensorflow backend [4].

For the SVM implementation, we used the SVM module of the Scikit-learn library for Python (Pedregosa et al., 2011), based on the C-support vector classification implementation of Chang and Lin (2011). We trained the model using a linear kernel and regularization parameter $C = 1$. For probabilistic prediction, the module uses Platt scaling (Platt et al., 1999), which is based on fitting a logistic regression on the initial binary scores using cross-validation.

5.3 Evaluation

A difficulty in evaluating on the Sami datasets is that the set of word pairs annotated as cognates in the Álgu database is known to be far from complete for the vocabularies covered. As a result, there are many word pairs in the vocabularies that are cognates, but are evaluated as unrelated. Measures such as accuracy and precision are therefore not useful for our problem setting, since we do not know whether a given word pair *not* among the annotated cognates is a cognate pair. We can, however, evaluate the *recall* of the known cognate pairs: what proportion of the annotated pairs make it into the set ranked as most likely cognates by the model. We use SAMI-FT-TEST to compute precision-recall curves for the fine-tuned S-CNN, unadapted S-CNN, SVM, and the baseline LD.

Computing scores for all pairs of words between two vocabularies is time consuming. Therefore, when evaluating on whole vocabularies, we only take those words q in vocabulary X that we know have at least one cognate in the other vocabulary Y. Then, we compute scores between each of these words, and all words in the other language. In order to evaluate these scores, we use recall@k averaged over the words q, the queries, and the set of language pairs in the test set. We call this metric the *mean average recall@k*:

$$\text{MAR@}k = \frac{1}{L}\sum_{l=1}^{L}\frac{1}{Q}\sum_{q=1}^{Q}\text{R@}k,$$

$$\text{where R@}k = \frac{\#\text{cognates within top-}k\text{ results}}{\#\text{cognates in } Y},$$

where Q is the number of queries, and L is the number of language pairs. That is, for each word query q, we rank the pairs $(q, y_i)\forall i$ and get the top 100 words y for each q. We then compute the recall@k for $k = 1, \ldots, 100$ for q, that is, counting the cognates found within the k highest-ranked words divided by the total number of cognates for q in Y. For most words q, there is only one, and for some there are several cognates in Y.

[2]The research group of Sami language technology at the University of Tromssa. http://giellatekno.uit.no/index.eng.html.

[3]Available at: http://kaino.kotus.fi/algu/

[4]Github repository available at: https://github.com/fchollet/keras.

5.4 Experimental setup

We present two experiments. We first train the SVM and S-CNN models on IE-TRAIN. (LD requires no training.)

In **experiment 1**, we apply these three models directly to the three pairs of Sami vocabularies (SAMI-FULL) to measure how well methods trained only on Indo-European data can identify cognates in Sami languages. This tells us how well the methods can exploit information from a different language family. In this experiment, we evaluate the models using the MAR@k metric (see section 5.3).

In **experiment 2**, we fine-tune the S-CNN on a small set of Sami cognates (SAMI-FT), containing example cognate pairs from all three language pairs. We test it on SAMI-FT-TEST to see how much of the performance loss from language transfer can be regained by providing the model with just a small amount of data from the target language family. We also analyse how the performance of the S-CNN improves with the amount of fine-tuning data it is given. In this experiment, we evaluate the models using precision-recall curves.

6 Results

6.1 Experiment 1: Indo-European models for Sami cognates

Figure 2 shows the MAR@k curves for Sami cognate identification for the three models trained on Indo-European data: S-CNN (without fine-tuning), SVM, and the baseline LD. The S-CNN outperforms the other approaches by a substantial margin, across values of k. This result suggests that the neural networks in the S-CNN are able to capture aspects of the cognateness relation that transfer across language families more effectively than the hand-designed features of the SVM. The SVM also outperforms LD – unsurprising, since the Levenshtein distance is included among its features.

Since the S-CNN performs best in this experiment, we use it in experiment 2, where we fine-tune the model on the target language family.

6.2 Experiment 2: Fine-tuning on target language family

Figure 3 shows how the number of cognate pairs used in fine-tuning improves average precision. Naturally, the average precision increases together with the number of cognates used in training. The improvement converges with about 500 training pairs, which is the number used for the fine-tuned model in Figure 4.

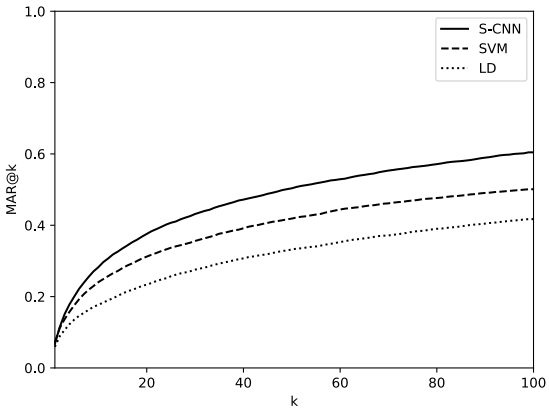

Figure 2: MAR@k for $k = 1 \ldots 100$, for SAMI-FULL, using models trained on IE-TRAIN. One curve is the average over all pairs of Sami languages.

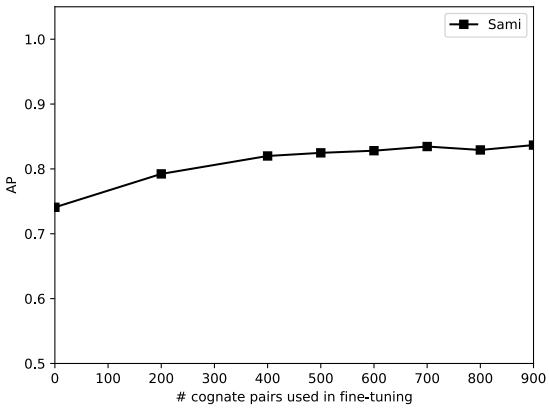

Figure 3: The learning curve of S-CNN fine-tuned on SAMI-FT, having been pre-trained on IE-TRAIN.

Figure 4 shows the precision-recall curves for each approach for the small-scale Sami test set (SAMI-FT-TEST). The corresponding values for average precision are given in Table 4. The pattern of results reflects that in Figure 2: the S-CNN outperforms the other two approaches based on string similarity metrics. The fine-tuned S-CNN substantially outperforms the untuned model. In terms of average precision, the improvement is approximately 11%.

This result tells us that, in addition to learning more general information about cognates that can be carried across language families than the SVM,

Approach	AP
S-CNN + FT	0.825
S-CNN	0.741
SVM	0.608
LD	0.540

Table 4: Average precision in the small-scale Sami test set for each approach.

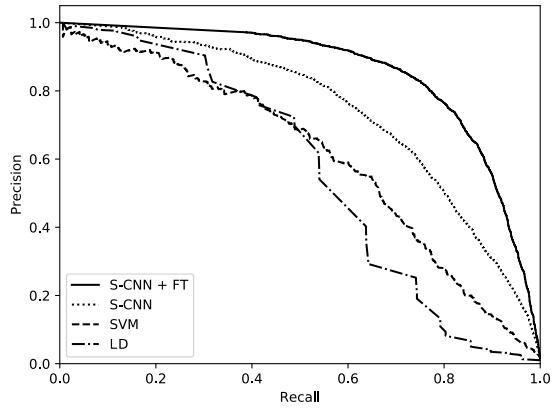

Figure 4: The precision-recall curves for each approach tested on SAMI-FT-TEST. S-CNN + FT was pre-trained on IE-TRAIN and fine-tuned on SAMI-FT. The unadapted S-CNN and SVM were trained on only IE-TRAIN.

the S-CNN is also able to make use of even a small number of annotated examples from the target languages to improve its predictions.

7 Conclusion and future work

We have addressed the problem of cognate identification within a set of three truly low-resource Sami languages of the Uralic family. We have examined the extent to which training data from a completely unrelated, higher-resource language family can be utilised for this task. We have taken two approaches to learn a similarity metric for cognateness from Indo-European etymological data, namely an SVM and an S-CNN, both applied to cognate identification in previous work, compared with a Levenshtein distance baseline. In addition, we have compared these with a fine-tuned S-CNN that has access to a small amount of training data in the target language family.

The results of our experiments have shown that the S-CNN is able to generalize more effectively across language families, compared with the SVM. Furthermore, a substantial improvement in performance can be attained by fine-tuning the model with only a small number of cognate examples from the target language set.

In future work, we will investigate whether language transfer for cognate identification can be further improved by making use of unsupervised multilingual character embeddings (Granroth-Wilding and Toivonen, 2019) instead of one-hot encoded characters. This could allow the model to exploit cross-lingual similarities in the usage patterns of symbols, replacing some of the manually encoded knowledge about correspondences across language pairs in previous work without the need to specify features by hand. In addition, due to the incomplete evaluation cognate sets, the experimental set-up could be complemented with a manual evaluation of top cognate suggestions in a manner similar to Hämäläinen and Rueter (2019).

Another avenue for future work is to investigate qualitatively how similar data-driven models generalize across other languages and language families, and how the choice of training language(s) affects performance. With such experimentation, we could gain more insight of what properties of sound change are carried over across families. In addition, we could investigate how the data-driven models presented here perform compared with models with more linguistically-informed handcrafted features.

Acknowledgements

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

References

Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2013. Cognate production using character-based machine translation. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 883–891.

Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2014. Readability for foreign language learning: The importance of cognates. *ITL-International Journal of Applied Linguistics*, 165(2):136–162.

Shane Bergsma and Grzegorz Kondrak. 2007. Alignment-based discriminative string similarity. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 656–663.

Michael Bloodgood and Benjamin Strauss. 2017. Using global constraints and reranking to improve cognates detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1983–1992, Vancouver, Canada. Association for Computational Linguistics.

Alexandre Bouchard-Côté, David Hall, Thomas L. Griffiths, and Dan Klein. 2013. Automated reconstruction of ancient languages using probabilistic models of sound change. *Proceedings of the National Academy of Sciences*, 110(11):4224–4229.

Remco Bouckaert, Philippe Lemey, Michael Dunn, Simon J. Greenhill, Alexander V. Alekseyenko, Alexei J. Drummond, Russell D. Gray, Marc A. Suchard, and Quentin D. Atkinson. 2012. Mapping the origins and expansion of the indo-european language family. *Science*, 337(6097):957–960.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27.

Will Chang, Chundra Cathcart, David Hall, and Andrew Garrett. 2015. Ancestry-constrained phylogenetic analysis supports the Indo-European steppe hypothesis. *Language*, 91(1):194–244.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 539–546. IEEE.

Alina Maria Ciobanu and Liviu P. Dinu. 2014. Automatic detection of cognates using orthographic alignment. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 99–105, Baltimore, Maryland. Association for Computational Linguistics.

Gerard de Melo. 2014. Etymological WordNet: Tracing the History of Words. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Luís Gomes and José Gabriel Pereira Lopes. 2011. Measuring spelling similarity for cognate identification. In *Progress in Artificial Intelligence*, pages 624–633, Berlin, Heidelberg. Springer Berlin Heidelberg.

Mark Granroth-Wilding and Hannu Toivonen. 2019. Unsupervised learning of cross-lingual symbol embeddings without parallel data. In *Proceedings of the Society for Computation in Linguistics*, volume 2, pages 19–28.

Stig-Arne Grönroos, Sami Virpioja, and Mikko Kurimo. 2018. Cognate-aware morphological segmentation for multilingual neural translation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 386–393, Belgium, Brussels. Association for Computational Linguistics.

Mika Hämäläinen and Jack Rueter. 2019. Finding Sami Cognates with a Character-Based NMT Approach. In *Proceedings of the Workshop on Computational Methods for Endangered Languages*, volume 1, pages 39–45.

Bradley Hauer and Grzegorz Kondrak. 2011. Clustering semantically equivalent words into cognate sets in multilingual lists. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 865–873, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Gerhard Jäger. 2014. Phylogenetic Inference from Word Lists Using Weighted Alignment with Empirically Determined Weights. In *Quantifying Language Dynamics*, pages 155–204. Brill, Leiden, the Netherlands.

Gerhard Jäger, Johann-Mattis List, and Pavel Sofroniev. 2017. Using support vector machines and state-of-the-art algorithms for phonetic alignment to identify cognates in multi-lingual wordlists. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1205–1216, Valencia, Spain. Association for Computational Linguistics.

Grzegorz Kondrak. 2000. A new algorithm for the alignment of phonetic sequences. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 288–295. Association for Computational Linguistics.

Grzegorz Kondrak. 2001. Identifying cognates by phonetic and semantic similarity. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics.

Grzegorz Kondrak. 2005. N-gram similarity and distance. In *International symposium on string processing and information retrieval*, pages 115–126. Springer.

Grzegorz Kondrak. 2009. Identification of Cognates and Recurrent Sound Correspondences in Word Lists. *Traitement Automatique des Langues (TAL)*, 50(2):201–235.

Vladimir I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics Doklady*, 10(8):707–710.

Johann-Mattis List. 2013. *Sequence comparison in historical linguistics*. Ph.D. thesis, Heinrich-Heine-Universität Düsseldorf.

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*, volume 1, pages 102–112.

Preslav Nakov and Jörg Tiedemann. 2012. Combining word-level and character-level models for machine translation between closely-related languages. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 301–305, Jeju Island, Korea. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

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*, pages 18–25. Association for Computational Linguistics.

Taraka Rama. 2015. Automatic cognate identification with gap-weighted string subsequences. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1227–1231, Denver, Colorado. Association for Computational Linguistics.

Taraka Rama. 2016. Siamese convolutional networks for cognate identification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1018–1027, Osaka, Japan. The COLING 2016 Organizing Committee.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Adam St Arnaud, David Beck, and Grzegorz Kondrak. 2017. Identifying cognate sets across dictionaries of related languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2519–2528, Copenhagen, Denmark. Association for Computational Linguistics.

Peter Turchin, Ilia Peiros, and Gell-Mann Murray. 2010. Analyzing genetic connections between languages by matching consonant classes. *Journal of Language Relationship*, 3:117–126.

Matthew D. Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28*, pages 649–657. Curran Associates, Inc.