Discovering Cognates Using LSTM Networks

Shantanu Kumar and **Ashwini Vaidya** and **Sumeet Agarwal**
Indian Institute of Technology Delhi
{ee1130798, ird11278, sumeet}@iitd.ac.in

Abstract

In this paper, we present a deep learning (DL) model for the task of pairwise cognate prediction. We use a character level model with recurrent neural network architecture and attention. We compare the performance of our model with previous approaches on various language families. We are able to show that our model performs better than non-DL methods which exploit surface similarity measures as well as a recent convolutional neural network (CNN) based model for the task.

1 Introduction

Cognates are words across different languages that are known to have originated from the same word in a common ancestral language. For example, the English word '*Night*' and the German word '*Nacht*', both meaning *Night* as well as the English '*Hound*' and German '*Hund*', meaning *Dog* are cognate word pairs, whose origin can be traced back to Proto-Germanic.

Traditionally, the identification of cognates was carried out by historical linguists, using word lists and establishing sound correspondences between words. These are useful in determining linguistic distance within a language family, and also to understand the process of language change. Cognate information has also been used in several downstream NLP tasks, like sentence alignment in bitexts (Simard et al., 1993) and improving statistical machine translation models (Kondrak et al., 2003). Additionally, it has been proposed that cognates can be used to share lexical resources among languages that are closely related (Singh and Surana, 2007).

For some time now, there has been a growing interest in automatic cognate identification techniques. Most approaches for this task focus on finding similarity measures between a pair of words such as orthographic or phonetic similarity (Hauer and Kondrak, 2011) (Inkpen et al., 2005) (List et al., 2016). These are used as features for a classifier to identify cognacy between a given word-pair. Surface similarity measures miss out on capturing generalizations beyond string similarity, as cognate words are not always revealingly similar. Rama (2015) attempt to identify cognates by looking at the common subsequences present in the candidate word pair. For a cognate pair like the English '*Wheel*' and the Sanskrit '*Chakra*', such an approach fails as there are no letters in common. In fact, even for a pair like English '*Father*' and Latin '*Pater*', a common subsequence approach completely ignores the similarity between the '*Fa*' and '*Pa*' phonemes, which is a possible indication of cognacy between the pair. Thus, there is a need of information about phonological similarity that is beyond surface similarity, such as the sound correspondences that are used in historical linguistics to narrow down candidate pairs as cognates.

By using DL based models, the need for external feature engineering is circumvented as the system learns to find hidden representations of the input depending on the task in hand. Our paper presents an end-to-end character-level recurrent neural network (RNN) based model that is adapted from a model used on a similar word-level task called RTE (Rocktäschel et al., 2016). Our model is able to outperform both the common subsequence model (Rama, 2015) as well as a recent CNN-based model (Rama, 2016) on the task.

2 Methods

The overall model used in our system is called the Recurrent Co-Attention Model (*CoAtt*). It is

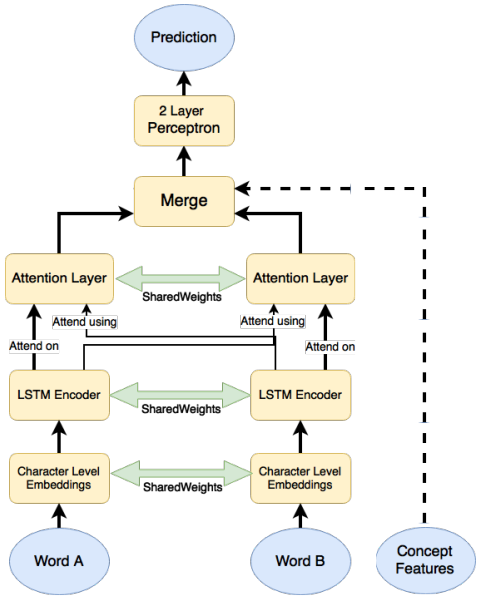

Figure 1: Recurrent Co-Attention Network for Cognate Discovery

adapted from the word-by-word attention model used by Rocktäschel et al. (2016) for the task of recognising textual entailment (RTE) in natural language sentences. Just as the RTE task involves understanding the semantics of a sentence which is hidden behind sequence of words, the cognate identification task also requires information beyond surface character similarity, which was the motivation to adapt this particular model for our task. The network is illustrated in Figure 1. We have converted the RTE model into a siamese-style network that encodes a word pair in parallel and then makes a discriminative judgement in the final layer.

The input words are first encoded into character level embeddings. Character embeddings are a form of distributional representation, where every character of the vocabulary is expressed as a vector in a vector space. Following the approach by Rama (2016), we define the character embeddings manually using various properties of the respective phoneme and then let these embeddings change during the training of the model, so that the representation is tuned for the task in hand. Initialising these embeddings to manually defined values rather than a random initialisation results in better training of the models.

The character encoded words are then further encoded using a bidirectional LSTM network. LSTMs or Long Short-Term Memory are a form of RNN and are extensively used in several NLP tasks. Finally the encoded words pass through a character by character attention layer. The attention layer essentially creates a weighted average of the sequence of vectors that represent a word coming out of the LSTM. The weights of the individual vectors in this average are computed dynamically depending on the characters present in the second word that it is being compared to. More intuitively, the attention mechanism helps to enhance the representation of a word obtained from the LSTM by giving it the 'context' of the second word. The attended encodings of both the words are merged and passed to a 2-layer perceptron with *tanh* and *sigmoid* activations to make a binary prediction.

Additionally, we also add a *Concept features* vector to the model by concatenating it with the merged attention vector before passing it to the 2-layer neural network. We hypothesise that information regarding the semantics or the meaning of the input word pair should be helpful for the classification task. The word semantics can provide information like the POS category of the word, which can be useful if some POS classes show higher degree of variation in cognates than others. We implement this by using GloVe word embeddings (Pennington et al., 2014). We use the GloVe embedding for the English concept of the word pair as obtained from the label in the dataset, and input this vector to the network.

3 Results

3.1 Experimental Setting

The task makes use of word lists of different language families taken from the basic vocabulary. A word list can be considered as a table where the different rows and columns represent different languages and concepts. Each cell in the table contains a lexical item along with its cognate class ID which helps to determine if two words are cognates. We worked with 3 language families in our experiments, namely Indo-European, Austronesian and Mayan. These families make a good test as they vary widely in terms of the number of languages, concepts and cognate classes.

We follow a Cross Language evaluation procedure, where the training and testing sample pairs are created using exclusive sets of languages. A random set of 70% of the languages is set as the training set of languages and the rest as testing set. Both words in a sample pair belongs to the same concept or meaning. It must be noted that cog-

Model	Indo-European		Austronesian		Mayan	
	F-Score	*AUC*	*F-Score*	*AUC*	*F-Score*	*AUC*
Gap-Weighted Subsequence	59.0	75.5	58.8	68.9	71.8	81.8
Phonetic CNN	73.7	86.1	54.6	68.0	72.8	85.0
Character CNN	75.3	85.3	62.2	71.6	75.9	85.7
LSTM + No Attention	56.7	59.0	51.2	55.2	60.6	67.1
LSTM + Uniform Attention	52.8	59.4	49.8	52.7	60.8	66.1
Co-Attention Model	83.8	89.2	69.0	77.5	67.1	67.7
+ IE	85.1	92.4	70.2	79.3	63.6	71.3
+ IE + CF	**86.2**	**93.0**	**70.5**	**79.7**	81.5	89.0
+ IE + PreT (Indo-European)	-	-	-	-	82.5	90.6
+ IE + PreT (Austronesian)	-	-	-	-	**83.5**	**91.2**

Table 1: Cross Language Evaluation Results
[IE: *Initialised Embeddings*, CF: *Concept Features*, PreT: *Pre-Training on another dataset*]

nate words are not simply the translations of each other, but are known to have a common origin historically.

3.2 Evaluation Metric

We report the *F-score* and the area under the PR curve (*AUC*) as a measure of performance for all the models. *F-score* is computed as the harmonic mean of the *precision* and *recall*[1]. Since the dataset is heavily biased and contains a majority of negative cognate sample pairs, we do not use *accuracy* as a measure of performance.

3.3 Baseline Models

We compare against several baseline models. *Gap-Weighted Subsequence* refers to the common subsequence model (Rama, 2015) mentioned earlier. The *Phonetic CNN* and *Character CNN* models are reimplementations of the CNN-based models (Rama, 2016). We also introduced two sanity-check baseline models to test the attention layer of the *CoAtt* model. The *LSTM + No Attention* model removes the Co-Attention layer from the *CoAtt* model, while the *LSTM + Uniform Attention* model does a simple average rather than a weighted average in the attention layer.

3.4 Experiments with Indo-European

We primarily worked with the Indo-European dataset on our experiments. As can be seen in Table 1, the *CoAtt* model is clearly an improvement over the CNN and the subsequence based models. The *LSTM + No Attention* and *LSTM + Uniform Attention* models reflect the importance of the attention layer adapted from the RTE model in the network, as without it the model does not perform very good.

A few additional features added to the *CoAtt* model helps to improve it even further. Initialising the character embeddings with the manually defined vectors (*IE* models) increases the *AUC* by around 3%. Further, addition of the *Concept features* discussed earlier, is also found to be useful.

3.5 Experiments with Multiple Datasets

We observe a similar trend for the models on the Austronesian and Mayan datasets as well. However, the *CoAtt* model does not train well on the Mayan dataset directly. This poor performance on the Mayan dataset is associated with its small size. The Mayan dataset being significantly smaller than the other datasets, does not prove sufficient for training the *CoAtt* network. We justify this hypothesis subsequently with the *Cross-Family Pretraining* experiment. The *Concept features* are again found useful to improve the *CoAtt* model, especially on the Mayan dataset, where the extra information about the meaning of input word pair helps the model to cross the baseline results.

Cross Family Pre-training Experiments

The three different language families with which we work have completely different origins and are placed across different regions geographically. We test if any notion of language evolution is still shared amongst these independently evolved families. This is done through the joint learning of models. The network is instantiated with the combined character vocabulary of two datasets. Then

[1]Precision and Recall is computed on positive labels at 0.5 threshold. Precision = TP/(TP+FP), Recall = TP/(TP+FN)

the model is trained on one dataset till the loss saturated. This is followed by the training on a second dataset, starting from the weights learned from the pre-training.

It is found that such a joint-training procedure helps the *CoAtt* model on the Mayan dataset significantly. The pretraining procedure is able to provide a good initialisation point to start training on the Mayan dataset. The pretrained models perform significantly better than the baseline models (*PreT* models in Table 1). This also provides evidence to support our hypothesis that the *CoAtt* was not able to learn on the Mayan dataset because of lack of enough data to train the network, but pre-training the model on other language families helped to show the true potential of the model on the dataset.

4 Discussion

The task of cognate discovery dwells into domain of finding rich hidden representation for words. It is found that simple surface similarity measures like common subsequence based features fail to capture the essence of phonological evolution and sound correspondences. Where there is large drift in the word structures and the characters of the words, these methods fail to capture any similarity between the words. Deep learning models like LSTMs are able to exploit hidden representations to make better judgments of word cognacy.

Cognate formation results from the evolution of sound changes in the words over time. From our experiments we have seen that there is a link in this evolution of sound class with the semantics of the words. Because words with different meanings are used in different frequencies, some appear to go through rapid adaptation and while others do not change by a lot. The models generally perform better on Nouns and Adjective words and they also have more number of cognate classes. In particular, concepts like *'WHAT'*, *'WHEN'*, *'HOW'* show a lot of variation even within a cognate class, so much that some cognate word pairs do not share any subsequence. Introducing concept features to the models in the form of word embeddings is seen to help in improving the results. It is also found that joint training of the models with data from different language families is also useful.

The task of discovering cognates can possibly be particularly useful among the languages of South Asia, which are not rich in lexical resources. Information about cognates can become an important source for assisting the creation and sharing of lexical resources between languages. By using DL models, the performance boosts are enough to test the model in an open domain. We applied our model to the domain of Hindi-Marathi, using a large unlabelled corpus of aligned texts to find cognate pairs and found through manual evaluation that the model is able to segregate the word pairs efficiently.

References

Bradley Hauer and Grzegorz Kondrak. 2011. Clustering semantically equivalent words into cognate sets in multilingual lists. In *IJCNLP*. Citeseer, pages 865–873.

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

Grzegorz Kondrak, Daniel Marcu, and Kevin Knight. 2003. Cognates can improve statistical translation models. In *Proceedings of the 2003 Conference of NAACL: Human Language Technologies*. ACL, pages 46–48.

Johann-Mattis List, Philippe Lopez, and Eric Bapteste. 2016. Using sequence similarity networks to identify partial cognates in multilingual wordlists. In *Proceedings of the ACL 2016*. pages 599–605. http://anthology.aclweb.org/P16-2097.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Taraka Rama. 2015. Automatic cognate identification with gap-weighted string subsequences. In *Proceedings of the 2015 Conference of NAACL: Human Language Technologies*. pages 1227–1231.

Taraka Rama. 2016. Siamese convolutional networks for cognate identification. In *Proceedings of COLING 2016*. pages 1018–1027.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomas Kocisky, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *ICLR*.

Michel Simard, George F Foster, and Pierre Isabelle. 1993. Using cognates to align sentences in bilingual corpora. In *Proceedings of the 1993 Conference of CASCON*. IBM Press, pages 1071–1082.

Anil Kumar Singh and Harshit Surana. 2007. Study of cognates among south asian languages for the purpose of building lexical resources. *Journal of Language Technology* .