Multilingual Transformer for cognate reflexes prediction

Alexandru Craevschi
University of Zurich
alexandru.craevschi@uzch.ch

Abstract

This paper addresses the problem of cognate reflexes prediction, focusing on the case study of the SIGTYP 2022 Shared Task. The objective of this task and its significance are initially introduced. Subsequently, a concise overview of the results obtained from the systems submitted for the SIGTYP Shared Task is provided. In contrast to the previously submitted systems, this paper proposes an alternative multilingual system. Although the multilingual Transformer[1] model does not achieve state-of-the-art results, the study demonstrates the potential of multilingual cognate reflexes prediction as a promising solution to this problem. Moreover, several improvements are suggested for enhancing the performance of the proposed system. By exploring the advantages of a multilingual approach, this research contributes to the advancement of cognate reflexes prediction and lays the groundwork for further advancements in this field.

1 Introduction

The task of cognate reflexes reconstruction is fundamental for historical linguistics. To understand what it consists of, it is important to understand the notion of *cognate* and of *reflex* separately. Cognate is "a word (or morpheme) that is related to a word (morpheme) in sister languages by reason of these words (morphemes) having been inherited by the related languages from a common word (morpheme) of the proto-language from which they descend" (Campbell and Mixco, 2007, p. 33). Cognates are often assembled in cognate sets, i.e., a group of words that were established to descend from a single ancestor word are clustered together. Reflex is a linguistic element (sound, construction, morpheme, etc.) that reflects the element of the proto-language from which it descends. Thus, individual cognates from languages under study could be considered reflexes of a proto-language. For example, English and German are members of a well-established Germanic subgroup of a bigger Indo-European language family. Words like German *zehn* and English *ten* descend from a proto-Germanic word for 'ten'.

To determine true cognates from mere coincidental similarities, linguists identify regular sound correspondences. For example, the grapheme <z> in German corresponds to <t> in English, as seen in words like German "Zahn" and English "tooth," or German "zu" and English "to." However, the task of establishing cognate reflexes goes beyond simple correspondences, as there are phonological conditioning factors to consider. These factors indicate that certain sounds may correspond to others only in specific phonological environments. Therefore, identifying accurate correspondences and establishing phonological conditioning requires careful attention to the preceding or subsequent sounds, in addition to potential trivial differences based on word position.

2 SIGTYP 2022 Shared Task

The SIGTYP 2022 Shared Task (List et al., 2022b) focused on reconstructing cognate reflexes across diverse language families using a subset of data from Lexibank (List et al., 2022a). The provided data consisted of cognate sets representing several language families, although not all families were fully represented, and some cognates were missing for some languages. For instance, the training data for the Indo-European family included only three Germanic languages and French. See Table 1 for an example of 2 cognate sets available as training data. The training data ranged from 771 to 10,139 words, while the surprise data had a range of 565 to 9,750 words, with a minimum of 4 and a maximum of 19 languages in each case. Additional details about the training and surprise data can be found in List et al. (2022b). The primary task objective was to predict a missing word in a cognate set based

[1] The code is available at https://github.com/acraevschi/cognate_reflexes_task

COGID	Dutch	English	French	German
423	s t eː n	s t əʊ n	–	ʃ t ai n
1521	b ɔ k	–	b u k	b ɔ k

Table 1: Two cognate sets available as training data. COGID is an ID of the meaning of a proto-form.

on the remaining words. However, even in the test setting, not all cognate sets were complete, as some words were either lost or not reconstructed by linguists. Five different training/testing proportions were used, varying from 10% to 50% of words retained for testing, thereby proportionally reducing the number of training samples. All the words were written in International Phonetic Alphabet (IPA).

3 Baseline and winning team

The task's baseline involved a preprocessing procedure specifically designed for computational historical linguistics. Then, two models were fitted. One model used Support Vector Machine (SVM), which showed stronger results than the alternative baseline in 10% proportion. The other baseline used the same preprocessing routine but instead the data were fitted using correspondence pattern recognition (CorPaR). This system was better out of the two baselines in 50% proportion. Among the systems, only two outperformed the baselines in the overall ranking, both developed by **Team Mockingbird** (Google Research).

Mockingbird's systems, described in (Kirov et al., 2022), featured a convolutional neural network (CNN) that emerged as the competition winner. Originally designed for restoring damaged pixels in images, the CNN was adapted to represent each language as a row and each character as a column in a sequence. The model will be referred to as model I1.

Another model presented by Mockingbird was a Transformer (Vaswani et al., 2017), a natural choice for the sequence-to-sequence prediction task at hand. However, the Transformer ranked second overall and required extensive data augmentation to achieve this result. The model employed three encoders: one for tokenized cognate words ("neighbor forms"), another for corresponding language names ("neighbor languages"), and a third for the target language name. The encoders shared embeddings for target words and neighbor words, while target language and neighbor languages shared a different embedding. The output of the encoders was concatenated and fed to the decoder. For additional details and information on augmentation techniques, refer to Kirov et al. (2022). These models will be referred to as N1-A[2] and N2. The difference lies in the fact that N1-A was trained using a development version of TensorFlow (Abadi et al., 2015) and Lingvo (Shen et al., 2019), while N2 was built using a public release. For the sake of space, I will only compare the results of my model to I1, N1-A, N2 and baselines.

4 Multilingual Transformer

All the models presented for the shared task were trained on datasets from individual families. The model to be introduced here is largely inspired by Mockingbird's Transformer with some notable differences. As the name suggests, the model is multilingual.[3] To ensure the compatibility between different sequence lengths of individual forms and different number of languages per family, the training data were padded up till maximal sequence length and to maximal number of neighbor languages per dataset (19 and 18 respectively). The maximal sequence length found in the data was 14 but 5 tokens were added to allow model to reconstruct sequences longer than the ones found in input.

The multilingual Transformer introduced here has some changes in the architecture as well. For instance, neighbor languages did not have a separate encoder. Neighbor language token was added to neighbor forms at the beginning of sequence. Thus, the model had two encoders: one for neighbor forms and one for target languages. Target language tokens were embedded. In case of neighbor forms, positional embeddings were used. To handle multiple input sequences, positional embeddings were computed for each sequence individually and then concatenated into a unified long sequence, which served as the input to the neighbor form encoder. The positional embeddings encompassed both token and absolute position embeddings, both with learnable parameters. Finally, the outputs of the two encoders were concatenated and served as input to the decoder, along with embedded target forms. Neighbor forms and target forms shared the same embedding. The total number of model's

[2] Mockingbird additionally presented N1-B and N1-C which differ from N1-A only by the number of steps they were trained for. Overall, N1-A was the best out of three.

[3] It is more accurate to say that it is a multi-family model but "multilingual" is a better known term in computational linguistics.

parameters is 5 million.

Hyperparameters proposed by Wu et al. (2021) were initially tested but found to be suboptimal for the current task. Specifically, the model was deemed overparameterized, and the batch size did not significantly affect the performance. The following hyperparameters were utilized in the model: the embedding dimension for target languages was set to 64, while the embedding dimension for forms was set to 192. A dropout rate of 0.15 was applied to both embeddings. The neighbor forms encoder consisted of 2 layers and 3 attention heads, with a feed-forward network (FFN) dimension of 256. A dropout rate of 0.1 was applied to this encoder. On the other hand, the target languages' encoder included only 1 layer with 4 attention heads and a FFN dimension of 512. It was discovered that a larger decoder configuration yielded better results. Hence, the decoder's FFN had a size of 1024, and there were 4 decoder layers with 5 attention heads. A dropout rate of 0.2 was applied to the decoder. The learning rate was set to decay exponentially. Two initial learning rates were tried: 0.001 and 0.0005. No significant difference was observed, but the latter was preferred by Keras Tuner and used in the final run. The model used Sparse Categorical Cross-Entropy as a loss function and Sparse Categorical Accuracy as metric. Early stopping criteria was applied after 5 consecutive epochs with no improvements for validation accuracy.

The model was finetuned using 5% of the training data. Importantly, 5% represent the proportion of cognate sets that were used as validation data. This allows us to avoid any leak between training and validation datasets, and capture overfitting in timely manner. Finetuning was performed using Keras Tuner (O'Malley et al., 2019). Afterwards, only 2.5% of the data were set as validation data. Given independence of validation data from training data, it was enough to stop training once the model started overfitting. Furthermore, the data was augmented, albeit very simplistically: the original data was copied and parts of it were removed to create a new sparser sample. An illustrated example of it is shown in Table 2. The number of forms to remove was random. Thus, if full cognate set had *N* forms and for sparse copy only *M* forms were retained, then another sparse cognate set had *N*-*M* forms. This results in model relying less on a concrete language when predicting the target sequence of another language and helps dealing with potential sparsity in test samples.

COGID	Dutch	English	French	German
423/full	s t eː n	s t əʊ n	–	ʃ t ai n
423/sparse	–	s t əʊ n	–	ʃ t ai n
423/sparse	s t eː n	–	–	ʃ t ai n

Table 2: Augmentation example of one cognate set. Last column is the target sequence and target language.

5 Results

After decoding the predicted sequences and storing the predictions in the format required by the organizers, I used official SIGTYP 2022 package which computed the performance of the model according to 4 metrics: edit distance, normalized edit distance (NED), B-Cubed F-scores, and BLEU. I will only report the last three metrics, as NED and edit distance represent the same metric normalized for words' length. Out of all the metrics, the reader might be unfamiliar with B-cubed F-score. List (2019) proposed a metric that only checks the regularity of occurrences between input and output sequences (List et al., 2022b). Thus, "if a method has systematic errors but otherwise does a good job in prediction, B-Cubed F-scores penalize results less strongly than edit distance". This metric is appropriate for the task, as the model should learn regular sound correspondences. The score varies from 0 to 1 with the higher values indicating better performance. Current model's results and results of other systems can be seen in Table 3. Note that the model was fitted separately on the whole training data and on the whole surprise data, as surprise data were meant only for an additional test. For setting with 50% proportion, no results were reported for N2 model but I show instead **CrossLingference-Julia** results. This is a non-neural system which was among best in B-cubed F-scores in scenario with scarce data.

Compared to other submissions, the model performed on par with the baseline when 50% of the data was removed. Furthermore, the multilingual transformer demonstrated superior performance compared to the public version of Mockingbird's N2 model, which was trained with extensive data augmentation. The authors noted that the number of samples with English as the target language increased significantly, from 300 to 4.2 million samples, representing a 14,000-fold increase in size. Consequently, the current workflow for the multilingual transformer may still have untapped po-

		Train data			Surprise data		
	SYSTEM	**NED↓**	**B-Cubed FS↑**	**BLEU↑**	**NED↓**	**B-Cubed FS↑**	**BLEU↑**
10% prop.	**Multiling. Transformer**	0.2920	0.6889	0.5893	0.3077	0.7191	0.5801
	Baseline-SVM	*0.2435*	*0.7435*	*0.6577*	0.2625	*0.7626*	0.6387
	Mockingbird-I1	**0.2255**	**0.7447**	**0.6805**	**0.2431**	**0.7673**	**0.6633**
	Mockingbird-N1-A	0.2674	0.7152	0.6316	*0.2568*	0.7604	*0.6479*
	Mockingbird-N2	0.2894	0.6823	0.6000	0.3135	0.7054	0.5744
50% prop.	**Multiling. Transformer**	0.3458	0.5877	0.5228	0.3956	0.5659	0.4701
	Baseline-CorPaR	0.3697	0.5978	0.5001	0.4445	0.5617	0.4265
	Mockingbird-I1	*0.3088*	*0.6307*	*0.5787*	**0.3518**	*0.6050*	**0.5337**
	Mockingbird-N1-A	0.3391	0.5907	0.5362	*0.3800*	0.5959	*0.4934*
	CrossLingference-Julia	**0.2993**	**0.6647**	**0.5914**	0.4274	**0.6193**	0.4296

Table 3: Comparison of results across four available data conditions. The results are averaged across all datasets included in the train data and surprise data. Two proportions, 10% and 50%, are considered. The best result is indicated in **bold**, and the second best is indicated in *italics*, based on the result per metric.

tential for achieving even better performance. It is worth noting that the Transformer model, being more complex than models like the SVM baseline or the CNN-based I1 model, demands greater computational resources. However, this demand is partially offset by the fact that the multilingual Transformer only needs to be trained once on all the data, unlike training multiple separate models for each language family. Surprisingly enough, despite multilingual model being larger than any of the two baselines, its performance does not drop as drastically when reducing the proportion of training data.

6 Discussion

The goal of this endeavor was to showcase the potential of multilingual transfer beyond machine translation, particularly in low-resource tasks like cognate reflexes reconstruction. In multilingual machine translation, it is common to share embeddings across languages to capture meta-language semantics. Some models employ language-general encoder/decoder layers along with a few language-specific layers (Fan et al., 2020). Conceptually, the idea of multilingual transfer in translation tasks is straightforward and relies on semantics. Although not all semantics are universal, there is a considerable degree of shared knowledge. When it comes to IPA tokens, it is likely that the model learns a small typology of sounds and their changes. Historical linguists rely on a set of typologically common sound changes when reconstructing a language. The model may have learned embeddings that cluster in a similar fashion as phonetic features, and through attention mechanisms, it can learn to identify phonological conditioning patterns. If one overcomes the current limitations of the model, it is practical to have a model pre-trained on large amount of already available data from a resource like Lexibank, and to then apply it in a few-short learning fashion to new language families. Such a model would be of great help to historical linguists in the long process of language reconstruction.

Limitations

While the model achieved decent results comparable to other submissions, it did not surpass the baseline in certain conditions. However, the model's current implementation is not optimized from a technical standpoint. It lacks key features found in other multilingual models, such as language-general and language-specific encoders/decoders. Additionally, relative positional encoding could enhance the model's ability to infer phonological environments compared to absolute position encoding. The use of augmented data, despite the increased computational cost, could potentially improve performance. Addressing the issue of unbalanced data is also important, as some language families have a large number of representatives and forms, while others have only a few. Notably, the model's performance remains poor when the dataset has a low number of languages, raising questions about its applicability to small language families. Furthermore, the task is currently performed in a supervised manner, whereas an ideal approach would be unsupervised, which presents a significant challenge yet to be tackled.

References

Martín Abadi, Ashish Agarwal, Paul Barham, et al. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Lyle Campbell and Mauricio Mixco. 2007. *A Glossary of Historical Linguistics*. Edinburgh University Press, Edinburgh.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, et al. 2020. Beyond english-centric multilingual machine translation. *CoRR*.

Christo Kirov, Richard Sproat, and Alexander Gutkin. 2022. Mockingbird at the sigtyp 2022 shared task: Two types of models for prediction of cognate reflexes. In *Proceedings of the 4th Workshop on Research in Computational Typology and Multilingual NLP (SIGTYP 2022) at NAACL*, pages 70–79, Seattle, WA. July, 2022.

Johann-Mattis List. 2019. Beyond edit distances: Comparing linguistic reconstruction systems. *Theoretical Linguistics*, 45(3-4):247–258.

Johann-Mattis List, Robert Forkel, Simon J. Greenhill, Christoph Rzymski, Johannes Englisch, and Russell D. Gray. 2022a. Lexibank, a public repository of standardized wordlists with computed phonological and lexical features. *Scientific Data*, pages 1–31.

Johann-Mattis List, Ekaterina Vylomova, Robert Forkel, Nathan Hill, and Ryan Cotterell. 2022b. The SIGTYP 2022 shared task on the prediction of cognate reflexes. In *Proceedings of the 4th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 52–62, Seattle, Washington. Association for Computational Linguistics.

Tom O'Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. 2019. Keras Tuner. `https://github.com/keras-team/keras-tuner`.

Jonathan Shen, Patrick Nguyen, Yonghui Wu, Zhifeng Chen, et al. 2019. Lingvo: a modular and scalable framework for sequence-to-sequence modeling. *CoRR*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*.

Shijie Wu, Ryan Cotterell, and Mans Hulden. 2021. Applying the transformer to character-level transduction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1901–1907, Online. Association for Computational Linguistics.