Transferring Learned Models of Morphological Analogy

Esteban Marquer[1,*], Pierre-Alexandre Murena[2] and Miguel Couceiro[1]

[1]Université de Lorraine, CNRS, LORIA, F-54000, France
[2]HIIT, Aalto University, Helsinki, Finland

Abstract
Analogical proportions are statements of the form "A is to B as C is to D", which have been extensively studied in morphology. Recent advances on learning models of analogy from quadruples pave the way for data-driven modeling and analysis of analogy. In morphology, recent work introduces a neural network classifier for morphological analogies (ANNc). In this paper, we study the transferability of ANNc across different axiomatic settings to show the importance of the data augmentation in the modeling of analogy. We also provide experimental results on transfer between two morphology datasets (Sigmorphon2016 and Sigmorphon2019) and between more than 27 languages to draw parallels between transfer performance and proximity between language families.

Keywords
Transfer, Morphological analogies, Analogy detection

1. Motivation and Context

The past decade has seen an increasing interest in analogical reasoning (AR) and of analogical proportions (APs), which are statements that four elements A, B, C, D are in analogy (usually written $A : B :: C : D$). Indeed, AR and APs are useful not only in the study of the mechanisms of human cognition [1] but also for applications in artificial intelligence [2, 3]. There are two basic tasks associated with AR: the first is *analogy detection* that corresponds to the task of deciding whether a quadruple A, B, C, D constitutes a valid AP, and the second is *analogy solving* that corresponds to finding the solution of an *analogical equation*, *i.e.*, an AP $A : B :: C : X$ where X is unknown.

Analogies between words and strings of symbols has long been studied [4, 5, 6, 7, 8, 9, 10] and makes for an experimental setting in which a wide range of analogies appear, from simple ($a : aa :: b : bb$) to more complex (*word*:*language*::*note*:*music*). In this paper, we focus on the study of morphological analogies, *i.e.*, analogies modeling changes of morphemes. In particular, we study how the deep learning approach for detecting morphological analogies proposed in [11, 12] behaves when transferred across domains. We call this approach Analogy Neural

ICCBR Analogies'22: Workshop on Analogies: from Theory to Applications at ICCBR-2022, September, 2022, Nancy, France
*Corresponding author.
esteban.marquer@loria.fr (E. Marquer); pierre-alexandre.murena@aalto.fi (P. Murena); miguel.couceiro@loria.fr (M. Couceiro)
0000-0003-2315-7732 (E. Marquer); 0000-0003-4586-9511 (P. Murena); 0000-0003-2316-7623 (M. Couceiro)

Network classifier (ANNc). Note that morphological transformations are linked to changes in the syntactic role of a word.

1.1. Axiomatic Setting

The notion of analogy is not consensual and there have been several efforts to create a common logical framework for AR that follow different axiomatic and logical approaches [6, 8]. For instance, Lepage [7] introduces the following 4 axioms in the linguistic context for analogical proportions: *symmetry* (if $A : B :: C : D$, then $C : D :: A : B$), *central permutation* (if $A : B :: C : D$, then $A : C :: B : D$), *strong inner reflexivity* (if $A : A :: C : D$, then $D = C$), and *strong reflexivity* (if $A : B :: A : D$, then $D = B$). While these axioms seem reasonable in the word domain, they can be criticized in other application domains [13]. A functional view of analogy is to consider a transformation f such that $B = f(A)$ and $D = f(C)$, resulting in analogies of the form $A : f(A) :: C : f(C)$ [5, 14]. As such, A and C may not be in the same conceptual domain as B and D. For example, A, C can be cities and B, D countries: *Marseille*:*France*::*Lyon*:*France* is an acceptable analogy (both Marseille and Lyon are cities of France). This is also an example where central permutation can be problematic, as it would give *Marseille*:*Lyon*::*France*:*France* which implies that Lyon and Marseille are identical due to strong inner reflexivity.

In this work, we use model transfer to compare how ANNc behaves when different sets of axioms are considered for training and evaluation. In particular, we consider axiomatic settings in which central permutation is accounted for differently, that we detail in Subsec. 2.2. The experimental results of this comparison are reported in Sec. 3.

1.2. Previous Work on Analogy Detection

The analogy detection task corresponds to classifying quadruples A, B, C, D into valid or invalid analogies. In other words, it can be seen as a binary classification task. Morphological analogy detection is used in the context of analogical grids [15], *i.e.*, matrices of transformations of various words, similar to paradigm tables in linguistics [16]. To detect analogies and build the analogical grids, Fam and Lepage [15] use the number of characters occurrences and the length of the longest common subword.

In the context of semantic word analogies, Bayoudh *et al.* [4] use Kolmogorov complexity as a distance measure between words for analogy detection, and Lim *et al.* [17] implement a data-driven approach. Using a dataset of semantic analogies, Lim *et al.* learn a neural network to classify quadruples A, B, C, D into valid or invalid analogies, using their embeddings e_A, e_B, e_C, and e_D. We adapt the latter approach to morphology by replacing the original GloVe [18] semantic embedding model by a character-level embedding model in our previous work [11], which significantly outperforms previous approaches.

Analogy detection and solving are closely related tasks. For instance, the morphological analogy solving approach of Langlais *et al.* [19] and the one of Murena *et al.* [14] are used as analogy classifiers in [11]: given a quadruple A, B, C, D, if D is in the predicted solutions of $A : B :: C : X$ then $A : B :: C : D$ is a valid analogy, otherwise it is invalid. Other works on solving analogies on character strings can be found in the literature, including Copycat by Hofstadter and Mitchel [5] but also works relying on embedding spaces [20, 17, 21, 9, 10]. For instance, the work of Lim *et al.* [17] also proposes a model for analogy solving in addition to ANNc. The former model was adapted to morphological word analogies in our previous work [21] and outperforms generative methods that do not rely on deep learning.

1.3. Model Transfer in Machine Learning

In this article, by transfer we mean applying a machine learning model on a *target* domain, different from the *source* domain used to train/learn the model. Different types of transfer are possible, from directly applying a model to the target model, to transferring the model and *finetuning* it on the target domain. The latter method is a type of *model adaptation*, which consists in altering a transferred model to fit the new data and which is a key step in the transfer methodology. It is also possible to transfer a part of a model and reuse it as a component of a larger model, as is usually done with large pretrained embedding models such as BERT [22], wav2vec2 [23], or vision transformers [24].

Transferring a model can serve two main purposes: achieving satisfying performance on the target domain while dealing with issues of the target domain or the model (lack of data or of labeled data, large training time, biases, *etc.*), or studying differences in the behavior of the model on different domains. In this work, we focus on the latter aspect.

1.4. Previous Experiments on Transferring ANNc

In [12], we performed multiple transfer experiments with ANNc on analogies extracted from Sigmorphon2016 [25] and Japanese Bigger Analogy Test Set [26] (which are now available in Siganalogies [27]). We transfered between languages to explore how the analogy model could generalize between domains, and built models on a subset of representative languages to determine the feasibility of a more general model of analogy. In both settings we obtained encouraging performance, but we were not able to fully explain the difference in performance between the languages used. We first experimented with what we called *full transfer*, in which all the components of the approach (character encoder, morphological embedding model and ANNc) are trained on the source language and transferred to the target language, without finetuning. This approach produced good overall results except on some languages using non-roman characters, and is the approach we take for the present article. To solve this *alphabet gap* issue, we used *partial transfer*, *i.e.*, the character encoder and morphological embedding model trained on the target language are reused instead of the ones trained on the source language. While this approach improved performance in case of *alphabet gap*, it was still far from the performance of models trained on the target language, probably due to a mismatch between the embedding model and the embedding space used by ANNc.

1.5. Our Contribution

In this paper, we introduce general elements of our experimental setting in Sec. 2. We extend the results of [12] in several ways:

- in Sec. 3, we use transfer to determine the impact of the axiomatic setting on the performance of the model, as mentioned above, and confirm that different training procedures results in compliance to different sets of axioms;
- in Sec. 4, we confirm that ANNc coupled with the morphological embedding of [11] generalizes to similar data, by transferring models of analogies in 8 languages between the Sigmorphon2016 and Sigmorphon2019 dataset;
- in Sec. 5, we leverage 42 high resource languages of Sigmorphon2019 [28] to extend previous results on inter-language transferability, and confirm previous hypotheses on the *alphabet gap* issue and the transferability of morphological analogies between related languages.

2. Datasets, Axiomatic Setting and Model Transfer

In this section we first present the analogical data. Then, we detail the default axiomatic setting CP (accepting central permutation as an axiom) and two variants $\overline{CP}$ (explicitly refusing central permutation) and $\neg CP$ (not taking central permutation into account). Finally, we specify our training, evaluation and transfer protocol.

2.1. Datasets

In our experiments we use the analogies available in Siganalogies [27], which are extracted from three datasets: Sigmorphon2016 [25], Japanese Bigger Analogy Test Set [26], and Sigmorphon2019 [28]. In Siganalogies the analogies are obtained by associating four words A, B, C, D, with B is a morphological transformation of A (*i.e.*, $B = f(A)$) and similarly $D = f(D)$, such that the morphological transformations are identical.

2.2. Data Augmentation and Axiomatic Setting

In previous works on ANNc, the model was trained using training examples obtained by permuting the four words of each analogy in the dataset. For the positive class, permutations of four words resulting in *valid* analogies (P^+) are generated from each analogy $A : B :: C : D$ in the dataset. For the negative class, permutations resulting in *invalid* analogies (P^-) are generated from each analogy of P^+, as they all are valid analogies.

In introduction, we mention the possibility of using different axiomatic settings, which leads us to experiment with *central permutation* among the most discussed axioms of analogical proportions [13] in Sec. 3. For this purpose, we consider three axiomatic settings (CP, $\neg CP$, and $\overline{CP}$) described below.

In the setting of [11, 12, 17], that we call CP, the axioms of [7] are used and central permutation is considered as an axiom for APs. Central permutation is thus used to generate the permutations in P^+_{CP}. In particular, the permutations in P^+_{CP} (Eq. (1)) can be obtained by applying successively *central permutation* and *symmetry*. Permutations that contradict these two axioms or *strong inner reflexivity* are used to obtain P^-_{CP}. Given a base form $A : B :: C : D$, P^+_{CP} and P^-_{CP} are as follows:

$$P^+_{CP} = \{\langle A, B, C, D\rangle, \langle C, D, A, B\rangle, \langle B, A, D, C\rangle, \langle D, C, B, A\rangle,$$

$$\langle A, C, B, D\rangle, \langle C, A, D, B\rangle, \langle B, D, A, C\rangle, \langle D, B, C, A\rangle\} \tag{1}$$

$$P^-_{CP} = \bigcup\nolimits_{\langle A', B', C', D'\rangle \in P^+_{CP}} \{\langle A', A', C', D'\rangle, \langle B', A', C', D'\rangle, \langle C', B', A', D'\rangle\} \tag{2}$$

We consider two settings in which central permutation is not an axiom: $\neg CP$ in which we discard the central permutation axiom, and $\overline{CP}$ in which we explicitly consider that applications of central permutation are invalid analogies. Discarding central permutation to obtain $\neg CP$ is the simplest way to refute the central permutation axiom, and results in the following sets of permutations:

$$P^+_{\neg CP} = \{\langle A, B, C, D\rangle, \langle C, D, A, B\rangle, \langle B, A, D, C\rangle, \langle D, C, B, A\rangle\} \tag{3}$$

$$P^-_{\neg CP} = \bigcup\nolimits_{\langle A', B', C', D'\rangle \in P^+_{\neg CP}} \{\langle A', A', C', D'\rangle, \langle B', A', C', D'\rangle\} \tag{4}$$

For $\overline{CP}$ we go one step further in refuting the central permutation axiom by considering that applications of central permutation are invalid analogies, as mentioned above. To do so, central permutation is removed from the valid permutations (as in $P^+_{\neg CP}$) and added to the permutations of the invalid class:

$$P^+_{\overline{CP}} = P^+_{\neg CP} \tag{5}$$

$$\begin{aligned} P^-_{\overline{CP}} = \bigcup\nolimits_{\langle A', B', C', D'\rangle \in P^+_{\overline{CP}}} \{&\langle A', A', C', D'\rangle, \langle B', A', C', D'\rangle, \langle B', A', C', D'\rangle, \\ &\langle A', C', B', D'\rangle\} \end{aligned} \tag{6}$$

In [11, 12] a subset of 8 permutations from P^-_{CP} is randomly sampled during training to obtain balanced classes (8 permutations of P^-_{CP} for 8 permutations of P^+_{CP}). For $\overline{CP}$ and $\neg CP$, to obtain balanced classes with a number of permutations comparable to CP, 8 permutations from each class are randomly sampled[1] during training.

2.3. Model Training and Evaluation and Transfer Method

The training of ANNc on the source domain is done in the same setting as [11], *i.e.*, using 5×10^4 analogies for the dataset and, for each, generating permutations as mentioned above to obtain 8 samples of the valid and 8 of the invalid class. Similarly, the testing is done on 5×10^4 base analogies and all the corresponding permutations. To evaluate the model, we use the *balanced accuracy*, *i.e.*, the average of the accuracy of the valid class and the accuracy of the invalid class, thus ignoring the number of permutations seen in each class. For the experiment on the axiomatic setting (Sec. 3) and on the transfer between datasets (Sec. 4) we use 10 random seeds to ensure stability across random initialization and random selection of the base analogies. For the experiment on the transfer between languages (Sec. 5), due to the large number experiments to perform, a single random seed is used.

In our experiments, the classifier and the embedding model with the character vocabulary of the source domain are transferred to the target domain, and the evaluation is performed without finetuning the models on the target domain. For simplicity, we write $source \rightarrow target$ to denote the transfer from the source domain to the target domain.

[1] If $n > 8$ permutations are available for the class, 8 different permutations are randomly selected. If $n < 8$ permutations are available, $8 - n$ randomly selected permutations are added, ensuring that each permutation appears at least once.

3. Impact of the Axiomatic Setting on the Classification Performance

The purpose of our first experiment is to study the impact of the training procedure of ANNc on the permutations it considers valid or invalid. To confirm that using different training procedures results in models that fit different axiomatic settings, we compare how models trained in the three settings described in Subsec. 2.2 (CP, $\neg CP$, and $\overline{CP}$) behave when transferred to the other settings.

3.1. Experimental Setup

We use the 11 languages of [11], *i.e.*, Sigmorphon2016 and Japanese Bigger Analogy Test Set, and transfer between the three axiomatic settings described in Subsec. 2.2: CP (using P^+_{CP} and P^-_{CP}), $\neg CP$ (using $P^+_{\neg CP}$ and $P^-_{\neg CP}$), and $\overline{CP}$ (using $P^+_{\overline{CP}}$ and $P^-_{\overline{CP}}$). Intuitively, the expected behavior is the following, also represented in the top left corner of Fig. 1:

- each model is expected to perform best on when the source setting is the same as the target setting ($CP \to CP$, $\neg CP \to \neg CP$, and $\overline{CP} \to \overline{CP}$);
- both $CP \to \overline{CP}$ and $\overline{CP} \to CP$ are expected to perform poorly, as the source and target settings are incompatible *w.r.t* CP;
- both CP and $\overline{CP}$ are expected to perform well on $\neg CP$, as the permutations in $\neg CP$ are common to both CP and $\overline{CP}$;
- the performances of $\neg CP \to CP$ and $\neg CP \to \overline{CP}$ are hard to predict, as $\neg CP$ has no constraints *w.r.t* CP.

3.2. Results and Discussion

The results of the experiment are reported in Fig. 1, and for all languages we observe the expected results with minor variations. First, on the target $\neg CP$ setting, all the models perform equally, instead of $\neg CP \to \neg CP$ performing slightly better. Second, the performance of $CP \to \overline{CP}$ and $\overline{CP} \to CP$ are not as low as expected, with the peculiarity that $CP \to \overline{CP}$ always outperforms $\overline{CP} \to CP$ by roughly 10%.

On the one hand, these results confirm that the training procedure does have an impact on which permutations will be considered valid or invalid by the model. On the other hand, the observed results match the expected Z shape for all languages, which supports the intuitions used to construct the expected results. As these intuitions rely on the differences between the axiomatic settings, this experiment confirm that models of a specific axiomatic setting can be obtained with the corresponding training procedure.

4. Transfer Between Morphological Datasets

The purpose of our second experiment is to check whether the model is able to generalize to a closely related domain with a slightly different distribution of morphological transformations. Indeed, for each language the morphological transformations in Sigmorphon2016 [25] and

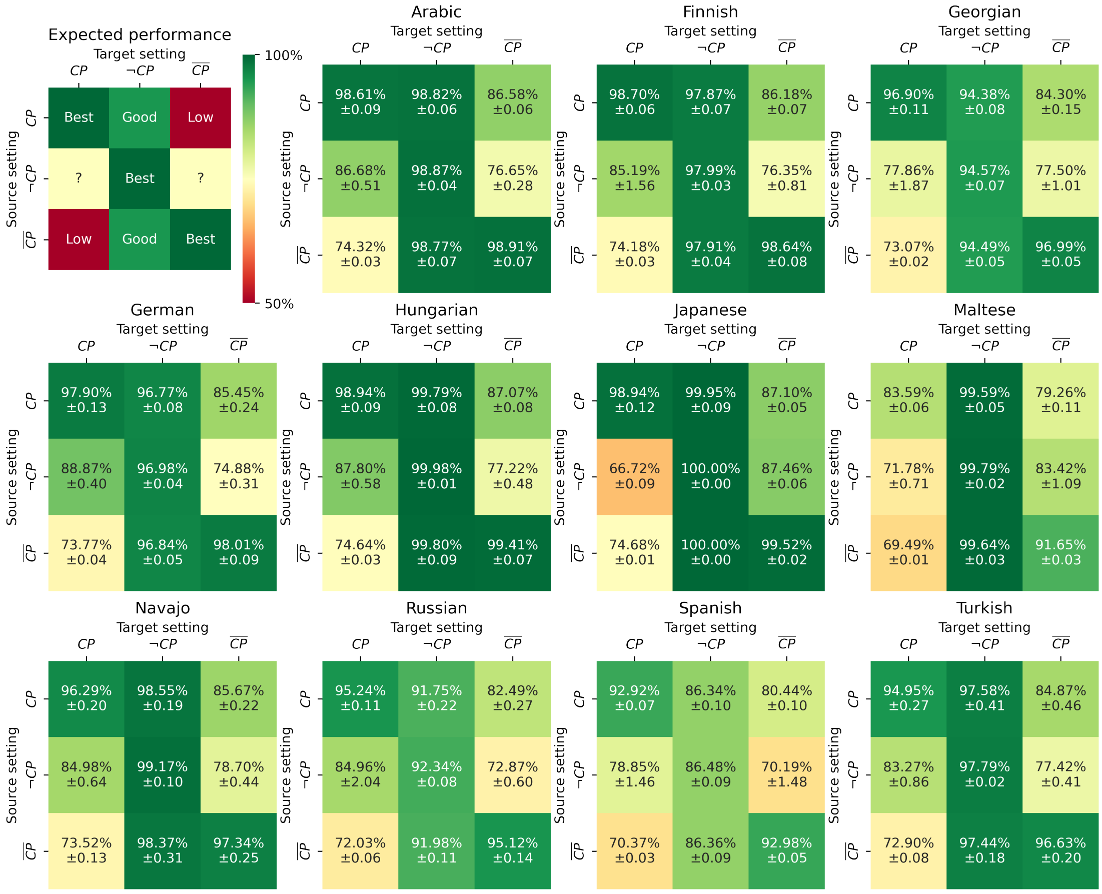

Figure 1: Balanced accuracy of, 10 per training setting. In the top left corner, a representation of the expected results

Sigmorphon2019 [28] are not exactly the same but the morphology of the language does not change.

4.1. Experimental Setup

For each language present in both Sigmorphon2016 and the high resource languages of Sigmorphon2019, we consider two transfer directions and two baselines:

- **19 → 16**: the transfer from the 2019 to the 2016 version of the language;
- **16** the baseline for 19 → 16: model trained and tested on Sigmorphon2016;
- **16 → 19**: the transfer from the 2016 to the 2019 version of the language;
- **19** the baseline for 16 → 19: model trained and tested on Sigmorphon2019.

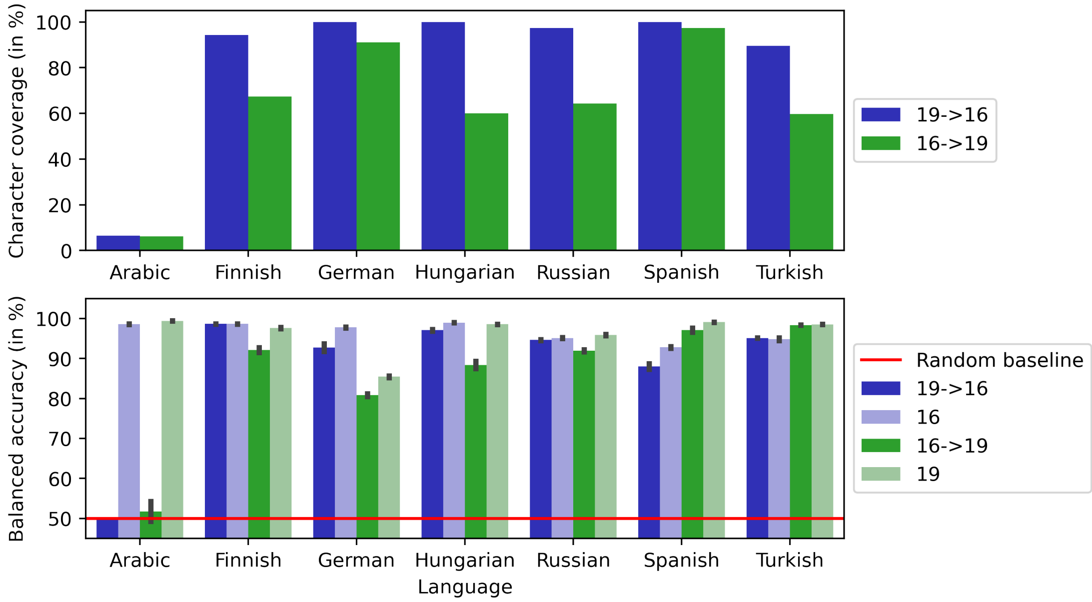

Figure 2: On the top: percentage of coverage of the target language characters by the source language characters. On the bottom: percentage of accuracy of the transferred model compared to the model trained on the target setting.

4.2. Results and Discussion

As shown in Fig. 2, the performance of the transferred model is comparable to or slightly lower than the non-transferred model. A significant correlation is noticeable between the performance of the transferred model and the character coverage of the target domains by the source domains (*i.e.*, the number of characters present in both domains divided by the number of characters present in the target domain), with a Pearson correlation coefficient of $r = 0.9639$ for $19 \to 16$ and $r = 0.7595$ for $16 \to 19$. When normalizing the transfer performance by the performance trained on the target domain, the correlation goes up to $r = 0.9739$ for $\frac{19\to16}{16}$ and $r = 0.8639$ for $\frac{16\to19}{19}$. A critical case of this correlation can be seen for Arabic, which is romanized in Sigmorphon2016 and not Sigmorphon2019, leading to a coverage close to 0%.

These results identify character coverage as a key factor in the transfer performance between strongly related domains. In this setup, the embedding model is the main source of performance loss.

5. Transfer Between Languages

To go beyond the limitations of character coverage, our third experiment leverages the large amount of multilingual data available in Sigmorphon2019 [28] to experiment following a similar intuition as in [12] but only between languages with similar alphabets.

5.1. Experimental Setup

We experiment with transfer between the high resource languages of Sigmorphon2019. We exclude Basque and Uzbek as they have less than 5×10^4 analogies. From the results of Sec. 4, we know that the amount of characters in common between the source and target domains strongly impacts the transfer performance. We extract clusters of languages sharing a significant part of their alphabets, and we transfer only within each cluster. This allows us to omit transfers likely to perform poorly due to the alphabet gap. We perform a total of 740 transfers, excluding cases where the source and target are the same. Compared to our other experiments, we reduce the number of test analogies from 5×10^4 to 5×10^3 and use a single random seed.

We use hierarchical clustering with the nearest point algorithm to get the clusters. Instead of the coverage between the source and the target language which is asymmetric, we use the Jaccard index[2] between the alphabets of the languages as a similarity measure. Using a threshold of 40% on the Jaccard and excluding singletons, we extract four clusters of at least two elements[3]. Based on our observations on the coverage[4], we consider relevant to include Romanian in both the Roman and Cyrilic clusters, and obtain the following language clusters named after the dominant alphabetic setting:

1. **Roman** cluster: Albanian, Asturian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Hungarian, Irish, Italian, Kurmanji, Latin, Latvian, Polish, Portuguese, Romanian, Slovak, Slovene, Sorani, Spanish, Swahili, Turkish, Welsh, and Zulu;
2. **Cyrillic** cluster: Adyghe, Bashkir, Belarusian, Bulgarian, Romanian, and Russian;
3. **Arabic** cluster: Arabic, Persian, and Urdu;
4. **Devanagari** cluster: Hindi and Sanskrit.

5.2. Results and Discussion

Once the languages with an overlap lower than 40% are eliminated, the Pearson correlation between the performance and the character coverage drops to $r = 0.6565$ for clusters 2, 3 and 4. For cluster 1, the largest cluster, coverage and performance appear uncorrelated with $r = 0.0379$. Similar values are observed when normalizing the transfer performance by the performance trained on the target domain. We report transfer performance for cluster 1 in Fig. 3.

We do not exclude that these correlations are influenced by the smaller amount of data used (only one seed, fewer testing analogies than usual). However, such a significant drop in correlation is unlikely if only this bias is involved. In fact, the tendencies we observe in the performance matrix indicate that the performance is linked to the language being used as a source language (*e.g.*, horizontal bar for English), and to the one being used as a target language (*e.g.*, vertical bars for Asturian and German). This behavior is likely due to either the quality of the learned model (how well it performs in general) or to the morphological similarities of some languages (at least within Sigmorphon2019). We exclude the former hypothesis, as only

[2]The Jaccard index between two finite sets A and B is $J(A, B) = \frac{A \cap B}{A \cup B}$.
[3]The corresponding dendrogram is provided in Appendix Fig. 2
[4]We provide the matrix of coverage for Sigmorphon2019 in appendix (Fig. 1).

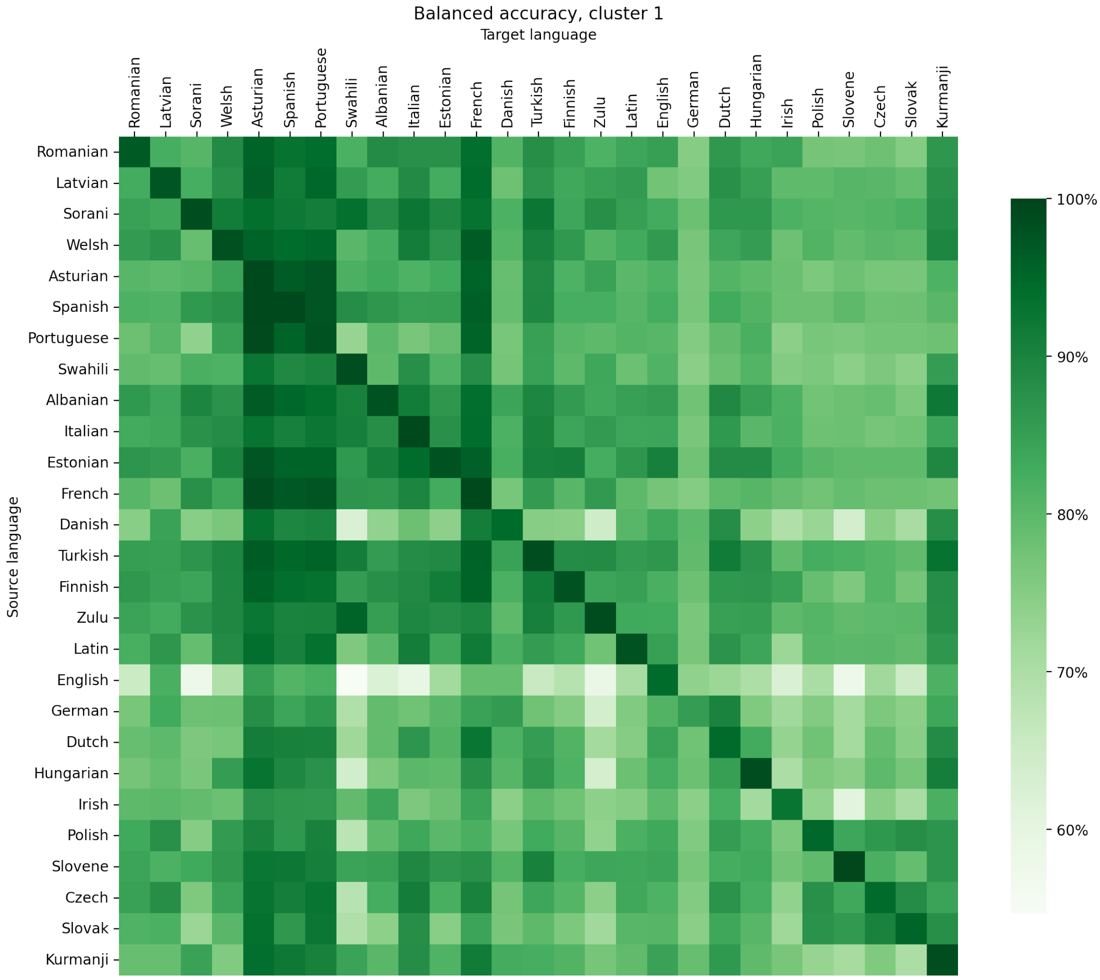

Figure 3: Transfer accuracy within cluster 1.

tendencies in the behavior as a source language (*i.e.*, horizontal bars) would be observed, while we mostly observe tendencies in the behavior as a target language (*i.e.*, vertical bars).

To confirm the influence of language similarities on performance, we explore hierarchical clustering within cluster 1 to study which key groups appear. When considering the behavior of the language as a target domain (*i.e.*, using performance from different source languages as a features for the clustering) rather than as a source domain, the clusters are more distinct. We focus on clusters extracted from the former, which can be seen in the dendrogram in Fig. 4. As a first analysis, we compare the corresponding clusters with language families as defined in Wikipedia. The Wikipedia page of each language contains a box with key information, the *infobox*. We use the "Language family" field of the infobox in the page of each language to determine how closely related they are, after minor corrections. The tree structure in Appendix Fig. 3, summarizes this information, with the leaves colored to match the colors of the clusters on Fig. 4. We find that the small clusters, which are the most easily distinguishable by the clustering

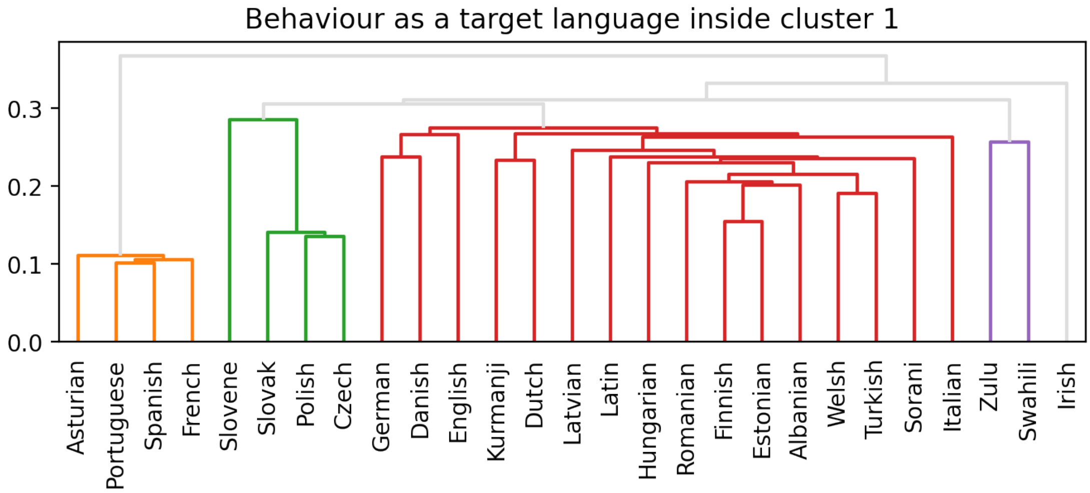

Figure 4: Dendrogram of the target languages, based on the transfer accuracy from all source languages as features for the target languages.

algorithm, correspond to closely related groups of languages. More precisely, the **orange** cluster contains Western Romance languages (Asturian, Portuguese, Spanish, and French), the **purple** cluster contains all the Bantu languages (Zulu and Swahili), and the **green** cluster contains Slavic languages: West Slavic languages (Slovak, Polish, and Czech) and slightly further the South Slavic language (Slovene). Finally, Irish is isolated and the **red** cluster contains all the remaining languages, even if distinct sub-clusters can be found: the (Finnish and Estonian) sub-cluster corresponds to Finnic languages and the (Romanian and Italian) sub-cluster contains the non-Western Romance languages. Other sub-clusters of the **red** cluster do not correspond to specific language families, like the (Kurmanji and Dutch) and the (Welsh and Dutch) sub-clusters.

From these results, we confirm that the morphological similarities of the languages are reflected in the model behavior during transfer. This indicates that our approach models morphological rules that can be transferred to related languages. However, it is clear that transfer in some clusters performs better than in others, though we are not yet able to provide explanations. Also, the performance is most likely influenced by the fact that the data in Sigmorphon2019 do not represent the full morphology of each language.

6. Conclusion

In this work, we use transfer to study the behavior of the ANNc analogy model when changing the axiomatic setting, the dataset, or the language of the analogies.

With results in 11 languages, we empirically confirmed that it is possible to model different axiomatic settings of analogy by changing the sets of permutations used when training ANNc. This highlights the importance of careful consideration on the axiomatic setting to use for data augmentation depending on the application, as it can significantly change model behavior. These results suggest that it is possible to determine the axiomatic setting matching a domain from data. Indeed, if domain data containing valid and invalid analogies is available, an ANNc model can be learned and matched against multiple axiomatic settings to find the one fitting the domain. This kind of method could provide empirical arguments to define the notion of analogy in specific domains.

We also extended previous results on tranferability between languages and complemented it with transferability between datasets. Empirical results confirm previous hypotheses on the alphabet gap issue. We found that in many cases it is possible to use the proximity in the Wikipedia language families to predict the performance of transferred models, which confirm the transferability of morphological analogies between languages. These results suggest that analogies and transfer could be used to empirically study morphological similarities between languages. Such similarities can be useful in language learning, by selecting languages known by a learner and having similar morphology to a language to learn. They could also be used to automatically create a data-driven language classification.

Acknowledgments

Experiments presented in this paper were carried out using computational clusters equipped with GPU from the Grid'5000 testbed (see https://www.grid5000.fr). This research was partially supported by TAILOR, a project funded by EU Horizon 2020 research and innovation program under GA No 952215, and the Inria Project Lab "Hybrid Approaches for Interpretable AI" (HyAIAI).

References

[1] M. Mitchell, Analogy making as a complex adaptive system, in: Santa Fe Institute Studies in the Sciences of Complexity, Reading, Mass.; Addison-Wesley; 1998, 2001, pp. 335–360.
[2] M. Mitchell, Abstraction and analogy-making in artificial intelligence, Ann. N.Y. Acad. Sci. 1505 (2021) 79–101.
[3] H. Prade, G. Richard, Analogical proportions: Why they are useful in ai, in: 13th IJCAI, Survey Track, 2021, pp. 4568–4576.
[4] M. Bayoudh, H. Prade, G. Richard, Evaluation of analogical proportions through kolmogorov complexity, Knowledge-Based Systems 29 (2012) 20–30.
[5] D. Hofstadter, M. Mitchell, The copycat project: A model of mental fluidity and analogy-making, in: Fluid Concepts and Creative Analogies, 1995, pp. 205–267.
[6] Y. Lepage, Analogy and formal languages, in: 6th CFG and 7th CML, volume 53, 2001, pp. 180–191.
[7] Y. Lepage, De l'analogie rendant compte de la commutation en linguistique, Habilitation à diriger des recherches, Université Joseph-Fourier - Grenoble I, 2003.
[8] L. Miclet, S. Bayoudh, A. Delhay, Analogical dissimilarity: Definition, algorithms and two experiments in machine learning, JAIR 32 (2008) 793–824.
[9] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: 1st ICLR, Workshop Track, 2013.

[10] T. Mikolov, W.-T. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in: NAACL, 2013, pp. 746–751.
[11] S. Alsaidi, A. Decker, P. Lay, E. Marquer, P.-A. Murena, M. Couceiro, A neural approach for detecting morphological analogies, in: IEEE 8th DSAA, 2021, pp. 1–10.
[12] S. Alsaidi, A. Decker, P. Lay, E. Marquer, P.-A. Murena, M. Couceiro, On the Transferability of Neural Models of Morphological Analogies, in: AIMLAI, ECML PKDD, volume 1524, 2021, pp. 76–89.
[13] C. Antic, Analogical proportions (2022).
[14] P.-A. Murena, M. Al-Ghossein, J.-L. Dessalles, A. Cornuéjols, Solving analogies on words based on minimal complexity transformation, in: 29th IJCAI, 2020, pp. 1848–1854.
[15] R. Fam, Y. Lepage, Tools for the production of analogical grids and a resource of n-gram analogical grids in 11 languages, in: 11th LREC, ELRA, 2018, pp. 1060–1066.
[16] R. Fam, Y. Lepage, Morphological predictability of unseen words using computational analogy., in: 24th ICCBR workshops, 2016, pp. 51–60.
[17] S. Lim, H. Prade, G. Richard, Solving word analogies: A machine learning perspective, in: 15th ECSQARU, volume 11726, 2019, pp. 238–250.
[18] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: EMNLP, 2014, pp. 1532–1543.
[19] P. Langlais, F. Yvon, P. Zweigenbaum, Improvements in analogical learning: Application to translating multi-terms of the medical domain, in: 12th EACL, ACL, 2009, pp. 487–495.
[20] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, R. Harshman, Using latent semantic analysis to improve access to textual information, in: SIGCHI, 1988, pp. 281–285.
[21] E. Marquer, S. Alsaidi, A. Decker, P.-A. Murena, M. Couceiro, A Deep Learning Approach to Solving Morphological Analogies, 2022. To appear in 30th ICCBR.
[22] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT, 2019, ACL, 2019, pp. 4171–4186.
[23] A. Baevski, Y. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: A framework for self-supervised learning of speech representations, in: NeurIPS, 2020, pp. 12449–12460.
[24] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: ICLR, OpenReview.net, 2021.
[25] R. Cotterell, C. Kirov, J. Sylak-Glassman, D. Yarowsky, J. Eisner, M. Hulden, The sigmorphon 2016 shared task–morphological reinflection, in: SIGMORPHON 2016, ACL, 2016, pp. 10–22.
[26] M. Karpinska, B. Li, A. Rogers, A. Drozd, Subcharacter information in japanese embeddings: when is it worth it?, in: RLSNA4NLP, ACL, 2018, pp. 28–37.
[27] E. Marquer, M. Couceiro, S. Alsaidi, A. Decker, Siganalogies - morphological analogies from Sigmorphon 2016 and 2019, 2022.
[28] A. D. McCarthy, E. Vylomova, S. Wu, C. Malaviya, L. Wolf-Sonkin, G. Nicolai, C. Kirov, M. Silfverberg, S. J. Mielke, J. Heinz, R. Cotterell, M. Hulden, The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection, in: 16th CRPPM workshops, ACL, 2019, pp. 229–244.

A. Coverage Between the Source and the Target Language

In Fig. 1, we can see the percentage of characters of a target language that are also present in the source language. In Fig. 2, we can see the dendrogram of the hierarchical clustering on the Jaccard index of the characters present in each pair of languages.

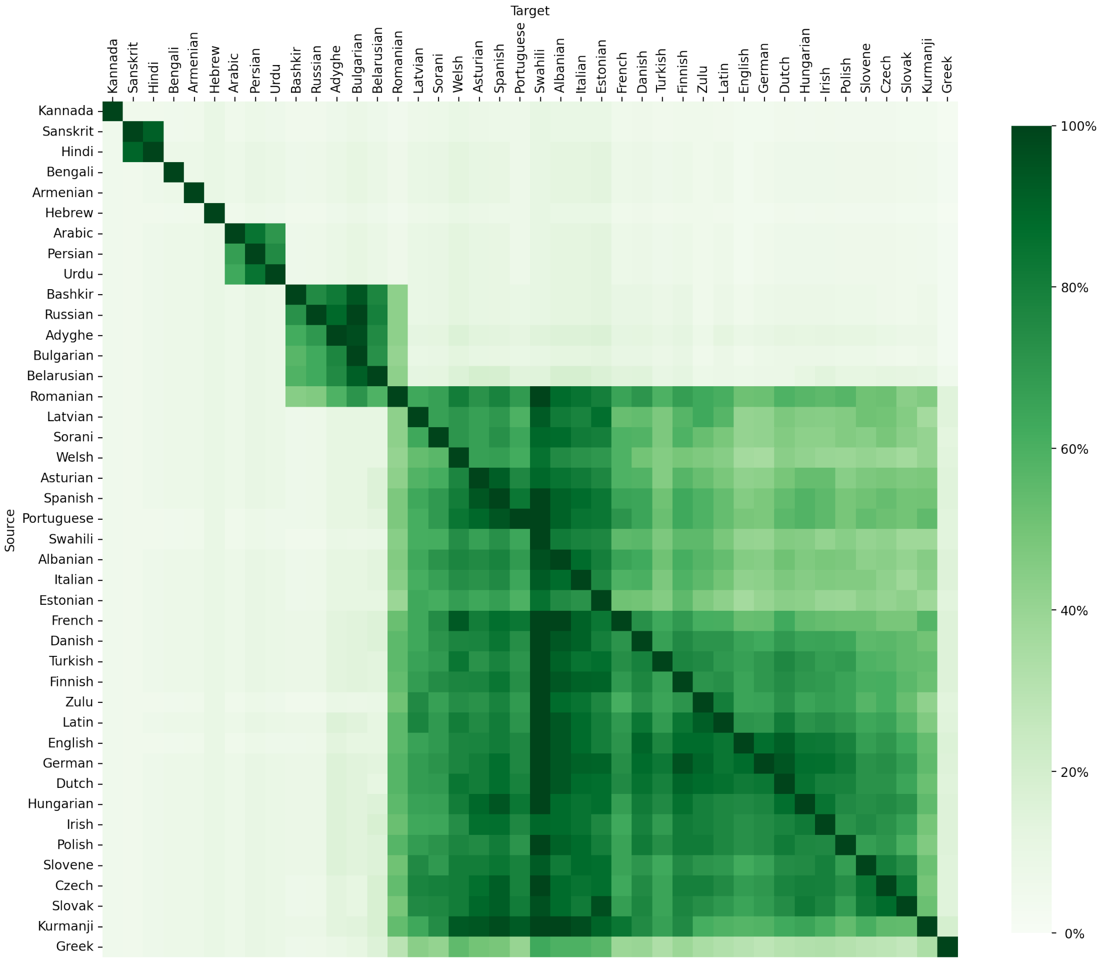

Figure 1: Coverage by the source language character vocabulary of the target language character vocabulary.

B. Transfer Performance Within the Largest Cluster of Languages

In Fig. 3, we can see the tree representing the language families of each of the languages in the lagest cluster (Albanian, Asturian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Hungarian, Irish, Italian, Kurmanji, Latin, Latvian, Polish, Portuguese, Romanian,

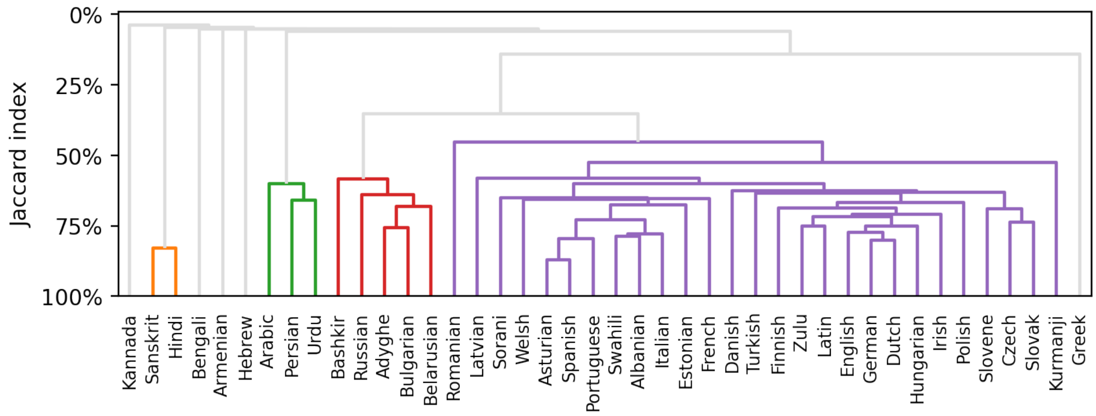

Figure 2: Dendrogram of the high resource languages in Sigmorphon2019 (except Basque and Uzbek), based on the Jaccard index between each pair of languages. With a threshold of 40% on the Jaccard and excluding singletons, four clusters (colored here) are found.

Slovak, Slovene, Sorani, Spanish, Swahili, Turkish, Welsh, and Zulu). The language families are extracted from the “Language family” field of the infobox in the Wikipedia page of each language.

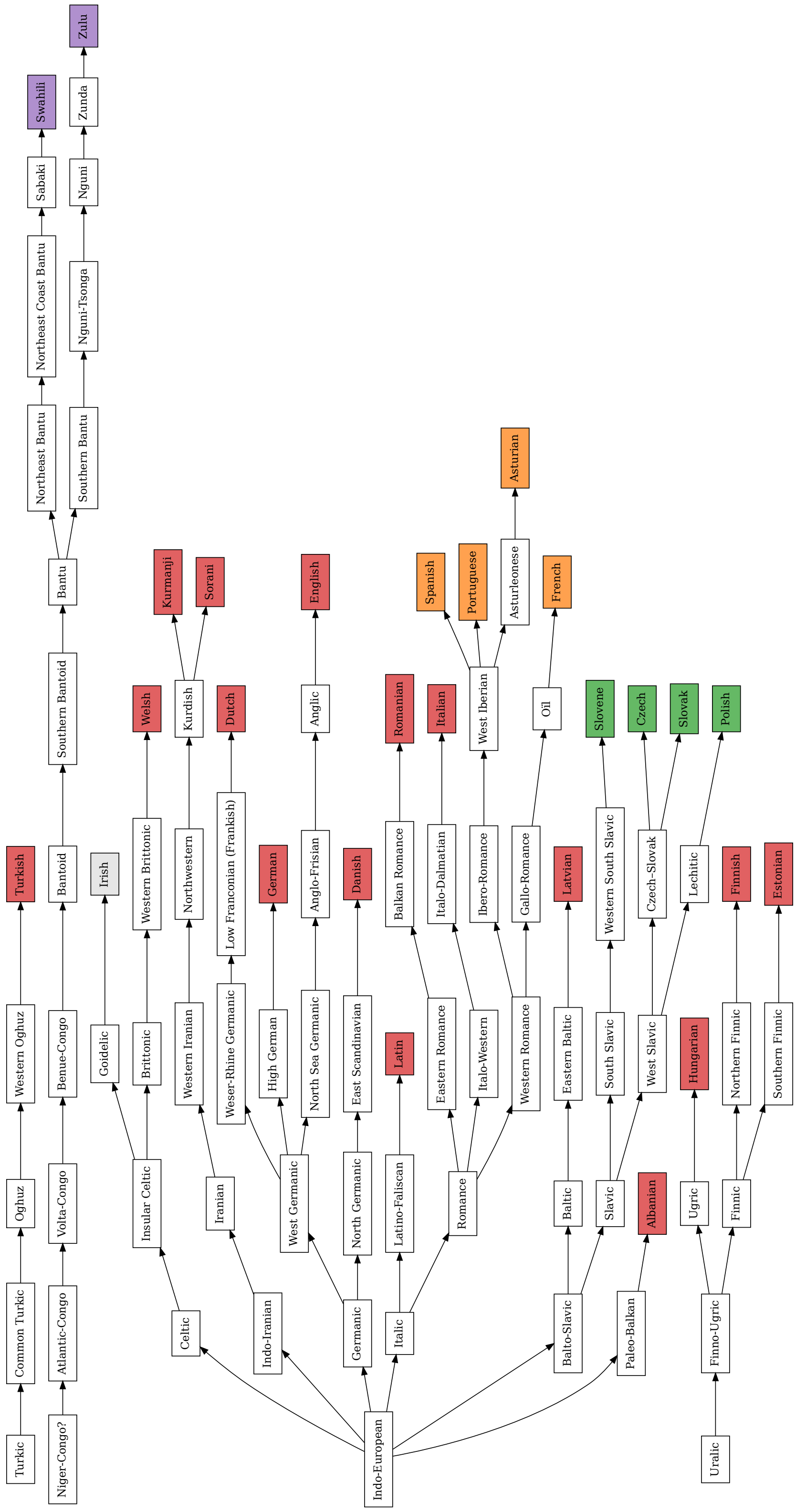

Figure 3: Trees of how the languages in cluster 1 relate based on their "Language family" according to Wikipedia. The Wikipedia page of each language contains an infobox (the area containing key information about the topic of the page), from which we extracted the "Language family" field. Languages are highlighted to match the clusters in the dendrogram of cluster 1.