Phylogenetic Multi-Lingual Dependency Parsing

Mathieu Dehouck
Univ. Lille, CNRS, UMR 9189 - CRIStAL
Magnet Team, Inria Lille
59650 Villeneuve d'Ascq, France
mathieu.dehouck@inria.fr

Pascal Denis
Magnet Team, Inria Lille
59650 Villeneuve d'Ascq, France
pascal.denis@inria.fr

Abstract

Languages evolve and diverge over time. Their evolutionary history is often depicted in the shape of a phylogenetic tree. Assuming parsing models are representations of their languages grammars, their evolution should follow a structure similar to that of the phylogenetic tree. In this paper, drawing inspiration from multi-task learning, we make use of the phylogenetic tree to guide the learning of multi-lingual dependency parsers leveraging languages structural similarities. Experiments on data from the Universal Dependency project show that phylogenetic training is beneficial to low resourced languages and to well furnished languages families. As a side product of phylogenetic training, our model is able to perform zero-shot parsing of previously unseen languages.

1 Introduction

Languages change and evolve over time. A community that spoke once a single language can be split geographically or politically, and if the separation is long enough their language will diverge in direction different enough so that at some point they might not be intelligible to each other. The most striking differences between related languages are often of lexical and phonological order but grammars also change over time.

Those divergent histories are often depicted in the shape of a tree in which related languages whose common history stopped earlier branch off higher than languages that have shared a longer common trajectory (Jäger, 2015). We hypothesize that building on this shared history is beneficial when learning dependency parsing models. We thus propose to use the phylogenetic structure to guide the training of multi-lingual graph-based neural dependency parsers that will tie parameters between languages according to their common history.

As our phylogenetic learning induces parsing models for every inner node in the phylogenetic tree, it can also perform zero-shot dependency parsing of unseen languages. Indeed, one can use the model of the lowest ancestor (in the tree) of a new language as an approximation of that language grammar.

We assess the potential of phylogenetic training with experiments on data from the Universal Dependencies project version 2.2. Our results show that parsers indeed benefit from this multi-lingual training regime as models trained with the phylogenetic tree outperform independently learned models. The results on zero-shot parsing show that a number of factors such as the genre of the data and the writing system have a significant impact on the quality of the analysis of an unseen language, with morphological analysis being of great help.

The remaining of this paper is organized as follows. Section 2 presents both the neural parsing model as well as the phylogenetic training procedure. Section 3 presents some experiments over data from UD 2.2. Section 4 presents some related works on multi-task learning and multilingual parsing. Finally, Section 5 closes the paper and gives some future perspectives.

2 Model

We propose a multi-task learning framework that shares information between tasks using a tree structure. The tree structure allows us to both share model parameters and training samples between related tasks. We instantiate it with a graph-based neural parser and use the language phylogenetic tree to guide the learning process, but it can in principle be used with any tree that encodes tasks relateness and any learning algorithm that supports fine-tuning.

In this section we first describe the intuition be-

Proceedings of NAACL-HLT 2019, pages 192–203
Minneapolis, Minnesota, June 2 - June 7, 2019.

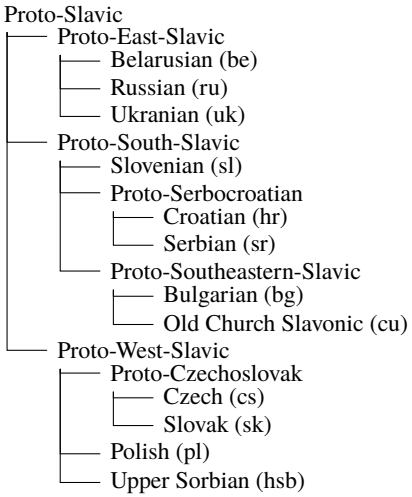

Figure 1: A possible phylogenetic tree for languages in the Slavic family.

hind phylogenetic training, then the neural parser and then the phylogenetic training itself.

2.1 Phylogenetic Hypothesis

Languages evolve from earlier stages and sometimes a language will change differently in different places leading to different languages with a common ancestor. This evolution process is often depicted in the shape of a tree in which leaves are actual languages and inner nodes can be either attested ancestral languages or their idealized reconstruction. Figure 1 gives an example of such a tree for a subset of the Slavic family of Indo-European languages (Simons and Fennig, 2018).

Just as languages evolve and diverge, so do their grammars. Assuming a parsing model is a parameterized representation of a grammar, then we can expect those models to evolve in a similar way. We thus take a multi-task approach to the problem of multi-lingual dependency parsing. What was once a single problem (e.g. parsing sentences in Proto-West-Slavic) becomes a set of distinct but related problems (parsing sentences in Czech, Polish, Slovak and Sorbian) as Proto-West-Slavic was evolving into its modern descendants.

We assume that the grammar of the last common ancestor is a good approximation of those languages grammars. Thus it should be easier to learn a language's grammar starting from its ancestor grammar than from scratch. There are however some issues with this assumption. First, a language grammar can be very different from its ancestor one from two millennia earlier. Consider the difference between modern French and early Classical Latin for example, in two millennia Latin has witnessed the loss of its case system and a complete refoundation of its verbal system. And the "last common" ancestors can have very different age depending on the languages we consider. We expect the common ancestor of Tagalog and Indonesian to be much much older than the common ancestor of Portuguese and Galician. Second, a lot of languages have only started to be recorded very recently thus lacking historical data all together. And when historical records are available, much work still needs to be done to render those data usable by parsers. For example the Universal Dependencies Project (Nivre et al., 2018) only has annotated corpora for Latin, old Greek, old Church Slavonic and Sanskrit. And even for those classical languages, it is not clear to which extent their modern counterparts really descend from them. Thus we need to find another way to access the ancestor language grammar than using historical data.

We propose to use all the data from descendent languages to represent an ancestor language. In principle, one could give more weight to older languages or to languages that are known to be more conservative, but this knowledge is not available for all languages families. Thus we resort to using all the available data from descendent languages without distinction.

Another problem is that the tree view is too simple to represent the complete range of phenomena involved in language evolution, such as language contacts. Furthermore, languages do not evolve completely randomly, but follow some linguistic universals and have to keep a balance between speakability, learnability and understandability. Thus, languages can share grammatical features without necessarily being genetically related, either by contact or by mere chance. However, the tree model is still a good starting point in practice and language families align well with grammatical similarity as recent works on typological analysis of UD treebanks have shown (Chen and Gerdes, 2017; Schluter and Agić, 2017). We thus make the simplifying assumption that a language grammar evolves only from an older stage and can be approximated by that previous stage.

2.2 Neural Model

Our scoring model is an edge factored graph-based neural model in the vein of recent works by Dozat et al. (Dozat et al., 2017). There are two major differences here compared to the parser of Dozat

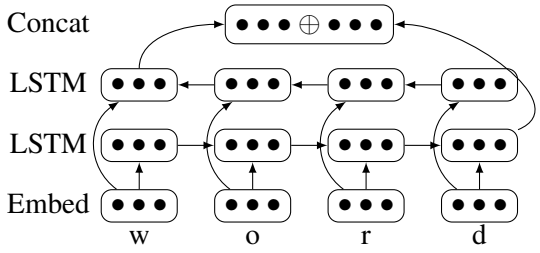

Figure 2: Bi-LSTM architecture for character based word representation. The final representation is the concatenation of the final cells of each layer.

et al. The first difference is in individual word representation, for which we use only the UPOS[1] tag, morphological information provided by UD treebanks and a character based word representation, whilst Dozat et al. use also the XPOS[2] tag, holistic word vectors (from Word2Vec (Mikolov et al., 2013) and their own) and they do not use morphological information beside what might already be given by the XPOS. The second difference is the scoring function proper. While they use biaffine scoring functions and decouple edge scoring from label scoring, we use a simple multi-layer perceptron to compute label scores and pick the max over the label as the edge score.

Let $x = (w_1 w_2 ... w_l)$ be a sentence of length l. Each word w_i is represented as the concatenation of 3 subvectors, one for its part-of-speech, one for its morphological attributes and one for its form:

$$\mathbf{w}_i = \mathbf{pos}_i \oplus \mathbf{morph}_i \oplus \mathbf{char}_i.$$

The part-of-speech vector ($\mathbf{pos}_i$) is from a look up table. The morphological vector ($\mathbf{morph}_i$) is the sum of the representation $\mathbf{m}_m$ of each morphological attribute m of the word given by the treebanks:

$$\mathbf{morph}_i = \sum_{m \in morph_i} \mathbf{m}_m.$$

We add a special dummy attribute representing the absence of morphological attributes.

The form vector ($\mathbf{char}_i$) is computed by a character BiLSTM (Hochreiter and Schmidhuber, 1997). Characters are fed one by one to the recurrent neural network in each direction. The actual form vector is then the concatenation of the outputs of the forward character LSTM and of the backward character LSTM as depicted in Figure 2.

[1]Universal part-of-speech for a set of 17 tags. Does not encode morphology.

[2]Language specific part-of-speech. Might include morphological information, but is not available for all languages.

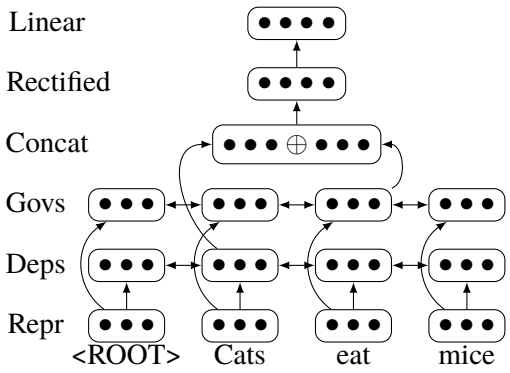

Figure 3: Neural network architecture for edge scoring. The contextualised representation of the governor (eat) and the dependent (Cats) are concatenated and passed through a rectified linear layer and a final plain linear layer to get a vector of label scores.

Once, each word has been given a representation in isolation, those representations are passed to two other BiLSTMs. Each word is then represented as the concatenation of its contextualised vector from the forward and backward layers:

$$\mathbf{c}_i = forward(\mathbf{w}_1, ..., \mathbf{w}_i) \oplus backward(\mathbf{w}_i, ..., \mathbf{w}_l).$$

We actually train two different BiLSTMs, one representing words as dependents ($\mathbf{c}$) and one words as governors ($\hat{\mathbf{c}}$). An edge score is then computed as follows. Its governor word vector $\hat{\mathbf{c}}_i$ and its dependent word vector $\mathbf{c}_j$ are concatenated and fed to a two layer perceptron (whose weights are $\mathbf{L}_1$ and $\mathbf{L}_2$) with a rectifier (noted $[...]^+$) after the first layer in order to compute the score s_{ijl} of the edge for every possible relation label l:

$$s_{ij} = \max_l s_{ijl} = \max_l(\mathbf{L}_2 \cdot [\mathbf{L}_1 \cdot (\hat{\mathbf{c}}_i \oplus \mathbf{c}_j)]^+)_l.$$

All the neural model parameters $\boldsymbol{\theta}$ (part-of-speech, character and morphological embeddings, character, dependant and governor BiLSTMs and the two layer perceptron weights) are trained end to end via back propagation one sentence at a time. Given a sentence x, we note j the index of the governor of w_i and l the relation label of w_i, the loss function is:

$$loss(x) = \sum_{w_i} \Big[\sum_{\substack{j' \neq j \\ j' \neq i}} \max(0, s_{ij'} - s_{ij} + 1)^2 + \sum_{l' \neq l} \max(0, s_{ijl'} - s_{ijl} + 1)^2 \Big]$$

For each word, there are two terms. The first term enforces that for all potential governors that are neither the word itself nor its actual governor, their highest score (irrespective of the relation label) should be smaller than the score of the actual governor and actual label by a margin of 1. The second term is similar and enforces that for the actual governor, any label that is not the true label should have a score smaller than the score of the actual label again by a margin of 1.

2.3 Phylogenetic Training

Let $\mathcal{L} = \{l_1, l_2, ..., l_{n_l}\}$ be a set of n_l languages and let $\mathcal{P} = \{p_1, p_2, ..., p_{n_p}\}$ be a set of n_p proto-languages (hypothesized ancestors of languages in $\mathcal{L}$). Let $\mathcal{T}$ be a tree over $\mathcal{L}^* = \mathcal{L} \cup \mathcal{P}$ such that languages of $\mathcal{L}$ are leaves and proto-languages of $\mathcal{P}$ are inner nodes. This means that we assume no two languages in $\mathcal{L}$ share a direct parenthood relation, but they at best descend both from a hypothesized parent. We could in principle have data appearing only in inner nodes. Tree $\mathcal{T}$ has a single root, a proto-language from $\mathcal{P}$ that all grammars descend from. This ancestor of all languages shall model linguistic universals[3] and ensure we deal with a well formed tree. We use the notation $p > l$ for the fact that language/node l descends from language/node p.

For each language $l \in \mathcal{L}$, we assume access to a set of n annotated examples $\mathcal{D}_l$. For each proto-language $p \in \mathcal{P}$, we create an annotated set $\mathcal{D}_p = \bigcup_{p>l} \mathcal{D}_l$ as the union of its descendent sets.

For each language $l \in \mathcal{L}^*$, we want to learn a parsing model $\boldsymbol{\theta}_l$.

2.3.1 Model Evolution

The main idea behind phylogenetic training is to initialize a new model with the model of its parent, thus effectively sharing information between languages and letting models diverged and specialize over time. The training pocedure is summarized in Algorithm 1.

At the beginning, we initialize a new blank/random model that will be the basic parsing model for all the world languages. Then, we sample sentences (we will discuss sampling issues in next section) randomly from all the available languages, parse them, compute the loss and update the model accordingly. Since the training sentences are sampled from all the available languages, the model will learn to be as good as possible for all the languages at the same time.

Data: a train set $\mathcal{D}_l$ and a dev set $\mathcal{D}'_l$ per language, a tree $\mathcal{T}$, two sampling sizes k, k' and a maximum number of reboot r
Result: a model $\boldsymbol{\theta}$ per node in $\mathcal{T}$
begin
  Instantiate empty queue Q
  $Q.push(\mathcal{T}.root)$
  while Q *is not empty* **do**
    $l = Q.pop()$
    if $l = \mathcal{T}.root$ **then**
      initialize $\boldsymbol{\theta}^0_{\mathcal{T}.root}$ randomly
    else
      $\boldsymbol{\theta}^0_l = \boldsymbol{\theta}_{l.parent}$
    $reboot = 0, i = 1, a_0 = 0$
    while $reboot < r$ **do**
      $\boldsymbol{\theta}^i_l = train(\boldsymbol{\theta}^{i-1}_l, \mathcal{D}_l, k)$
      $a_i = test(\boldsymbol{\theta}^i_l, \mathcal{D}'_l, k')$
      if $a_i \leq a_{i-1}$ **then**
        $reboot \ += 1$
      else
        $reboot = 0, i \ += 1$
    $\boldsymbol{\theta}_l = \boldsymbol{\theta}^i_l$
    for c *in l.children* **do**
      $Q.push(c)$

Algorithm 1: Phylogenetic training procedure.

When the model $\boldsymbol{\theta}_p$ has reached an optimum (that we defined hereafter), we pass a copy of it to each of its children. Thus, for each child c of p, we initialize $\boldsymbol{\theta}^0_c = \boldsymbol{\theta}_p$ to its parent (p) final state. Each model $\boldsymbol{\theta}_c$ is then refined on its own data set $\mathcal{D}_c$ which is a subset of $\mathcal{D}_p$, until it reaches its own optimum state and is passed down to its own children. This process is repeated until the model reaches a leaf language, where the model $\boldsymbol{\theta}_c$ is eventually refined over its mono-lingual data set $\mathcal{D}_c$.

By passing down optimal models from older/larger languages sets to newer/smaller ones, models get the chance to learn relevant information from many different languages while specializing as time goes by.

The question now is to find when to pass down a model to its children. In other words, at which stage has a model learned the most it could from its data and should start to diverge to improve again?

Following the principle of cross-validation, we

[3] It does not imply anything about our belief or not in the monoglotto genesis hypothesis.

propose to let held-out data decide when is the right time to pass the model down. Let $\mathcal{D}'_p$ be a set of held-out sentences from the same languages as $\mathcal{D}_p$. Then, after every epoch i of k training examples, we freeze the model $\boldsymbol{\theta}_p^i$, and test it on k' sentences from $\mathcal{D}'_p$. This gives a score a_i (UAS/LAS) to the current model. If the score is higher than the score of the previous model $\boldsymbol{\theta}_p^{i-1}$ then training goes on, otherwise we discard it and retrain $\boldsymbol{\theta}_p^{i-1}$ for another k sentences. If after having discarded r epochs in a raw we have not yet found a better one, then we assume we have reached an optimal model $\boldsymbol{\theta}_p^{i-1}$ and pass it on to its children (unless it is a leaf, in which training is over for that language).

2.3.2 Sentence Sampling

There are a few things we should consider when drawing examples from a proto-language distribution. Beside the question of whether some languages are more conservative than others with respect to their ancestor, which we have decided to simplify saying that all languages are as representative of their ancestors, there is the problem of data unbalance and tree unbalance.

Sampling sentences uniformly across languages is not a viable option for the size of datasets varies a lot across languages and that they do not correlate with how close a language is to its ancestor. For example, there are 260 Belarusian training sentences against 48814 Russian ones. The basic question is thus whether one should draw examples from languages or branches. Basic linguistic intuition tells us that drawing should be performed on branches. Modern languages distribution has no reason to reflect their proximity to their ancestor language. Amongst Indo-European languages, there are one or two Armenian languages as well as one or two Albanian languages (depending on the criteria for being a language), while there are tens of Slavic languages and Romance languages. However, there is no reason to believe that Slavic or Romance languages are better witnesses of proto-Indo-European than Armenian or Albanian.

Drawing examples from languages would bias the intermediate models toward families that have more languages (or more treebanks). It might be a good bias depending on the way one compute the overall accuracy of the system. If one uses the macro-average of the individual language parsers, then biasing models toward families with many members should improve the accuracy overall.

In this work, at a given inner node, we decided to sample uniformly at random over branches spanning from this node, then uniformly at random over languages and then uniformly at random over sentences. It boils down to flattening the subtree below an inner node to have a maximum depth of 2. For example in Figure 1, at the root (Proto-Slavic) we pick a branch at random (e.g. Proto-South-Slavic), then a language at random (e.g. Croatian) then a sentence at random. Given that we have picked the Proto-South-Slavic branch, all South-Slavic languages are then as likely to be chosen. This biases slightly the model toward bigger sub-families. In our example, Croatian and Serbian have the same chances to be sampled than Slovenian, therefore their family, Proto-Serbocroatian is twice as likely to be chosen as Slovenian is, while being at the same depth in the tree.

We could otherwise sample over branches, then over sub-branches again and again until we reach a leaf and only then pick a sentence. In this case, Proto-Serbocroatian and Slovenian would have the same probability to be chosen. This would give much more weight to languages high in the tree than languages low in the tree. While this would give more balance to the actual model, it could be detrimental to the averaged results since the data distribution is itself unbalanced. It would of course be possible to try any variation between those two, picking sub-branches according to a probability that would depend on the number of languages in that family for example, therefore mitigating the unbalance problem.

2.4 Zero-Shot Parsing

An interesting property of the phylogenetic training procedure is that it provides a model for each inner node of the tree and thus each intermediary grammar. If one were to bring a new language with its position in the tree, then we can use the pre-trained model of its direct ancestor as an initialization instead of learning a new model from scratch. Similarly, one can use this ancestor model directly to parse the new language, effectively performing zero-shot dependency parsing. We investigate this possibility in the experiment section.

3 Experiments

To assess the potential of phylogenetic training both in terms of multi-task learning and zero-shot parsing capabilities, we experimented with data from the Universal Dependencies project version

2.2 (Nivre et al., 2018). When several corpora are available for a language, we chose one to keep a good balance between morphological annotation and number of sentences. For example, the Portuguese GSD treebank has slightly more sentences than the Bosque treebank but it is not well morphologically annotated. The zero-shot parsing models have been directly tested on languages that lack of training set. The treebanks names are given in the tree 4 and the result table 1.

3.1 Setting

As some languages have no training data and unique writing systems making the character model inefficient for them, we resorted to use gold parts-of-speech and morphological attributes rather than predicted ones. For example, Thai has no training data, no language relative and a unique script, which altogether make it really hard to parse (from a phylogenetic perspective).

The phylogenetic tree used for the experiment is adapted from the Ethnologue (2018). For space reasons, it is reported in the appendix in Figures 4 and 5. We tried to have a tree as consensual as possible, but there are still a few disputable choices, mostly about granularity and consistency. Sanskrit could have its own branch in the Indic family just as Latin in the Romance family, but because Sanskrit has no training data, that would not actually change the results. Likewise, as Czechoslovak and Dutch-Afrikaans have their own branches, Scandinavian languages could also distributed between east and west Scandinavian. As an English based Creole, Naija could as well be put in the Germanic family, but we kept it as a separate (Creole) family.

Regarding model training proper, we used $k = 500$ training sentences per iteration, $k' = 500$ held-out sentences from the developpement set to compute running LAS and a maximum number of reboot $r = 5$. Following Dozat et al (2017), we use Eisner algorithm (Eisner, 1996) at test time to ensure outputs are well formed trees. The neural model is implemented in Dynet (Neubig et al., 2017) and we use Adadelta with default parameters as our trainer. We averaged the results over 5 random initializations. Independent models are trained in the same manner but with mono-lingual data only. We report both labeled and unlabeled edge prediction accuracy (UAS/LAS). In the appendix we also report results averaged per family.

3.2 Multi-Task Learning

Table 1 reports parsing results for languages that have a training set. Note that a few languages do not have a separate developpement set, then we used the training set for both training and validation. The training set size of those languages is reported in square brackets. This has low to no impact on other languages results but it can be problematic for the language itself as it can over-fit its training data especially when they are very few as is the case of Buryat for example. To be fair, we report two different averages. Avg is the average over languages that have a separate developpement set, and Avg No Dev is the average over languages that do not have a separate developpement set. For each language, the best UAS/LAS are reported in bold.

On average, phylogenetic training improves parsing accuracy, both labeled and unlabeled. This is especially true for languages that have very small training sets (50 sentences or less) and lack of developpement set. Those languages show an averaged 7 points improvement and up to 15 points (hsb, kmr). Since independent mono-lingual models follow the exact same training procedure but without phylogenetic initialization and that every sentence will be seen several times both at training and validation, the sampling method cannot explain such a difference. This shows that the ancestor's model is a good initialization and acts as a form of regularization, slowing down over-fitting.

Phylogenetic training is also beneficial as one gains information from related languages. Indo-European languages gain from sharing information. This is especially true for Balto-Slavic (sk +5.82, lt +5.07 UAS) and Indo-Iranian languages (mr +2.05 UAS). It is less consistent for Romance and Germanic languages. This might be due to the tree not representing well typology for those families. Typically, English tends to group syntactically with Scandinavian languages more than with West-Germanic. Turkic and Uralic languages show the same benefits overall (ug +2.67, fi +3.39 UAS).

Dravidian and Afro-Asiatic languages are not as consistent. While Telugu seems to gain from Tamil data, the reverse is not true. Result variation for Arabic, Hebrew and Coptic are marginal. This is likely due to the fact that we only have three quite different languages from that family and that they all have their own script.

Similarly, phylogenetic training is not consistently useful for languages that do not have rela-

	Phylogenetic		Independent	
	UAS	LAS	UAS	LAS
ar nuyad	74.81	70.32	**75.07**	**71.08**
cop	85.51	79.28	**86.03**	**80.15**
he	**81.89**	75.36	81.59	**75.57**
bxr [19]	**48.72**	**30.68**	37.88	18.09
eu	76.81	69.51	**78.61**	**72.76**
af	85.15	80.94	**85.44**	**81.66**
da	78.50	72.50	**79.16**	**74.13**
de gsd	**80.37**	**73.54**	79.48	72.37
en ewt	79.25	74.34	**79.27**	**74.66**
got	77.83	71.54	**79.91**	**74.33**
nb	**84.62**	**78.78**	83.82	78.09
nl alpino	**77.19**	**68.55**	76.52	68.40
nn nynorsk	82.39	76.44	**82.58**	**77.32**
sv talbanken	80.46	74.62	**81.17**	**75.47**
be	**80.18**	**74.11**	78.09	72.76
bg	86.01	79.16	**86.40**	**79.79**
cs pdt	**79.78**	**71.71**	77.45	69.88
cu	82.98	77.19	**83.31**	**78.32**
hr	**81.70**	**74.73**	81.05	73.95
hsb [23]	**74.24**	**66.01**	58.59	50.37
lt	**61.42**	**50.88**	56.35	46.14
lv	**78.39**	**70.14**	76.69	68.89
pl lfg	**92.88**	**88.53**	91.07	86.49
ru syntagrus	**77.91**	72.72	77.33	**72.85**
sk	**84.91**	**79.17**	79.09	73.20
sl ssj	87.15	83.43	**88.39**	**85.21**
sr	85.85	79.86	**86.17**	**80.47**
uk	**78.16**	**73.50**	74.96	70.91
ca	84.67	78.81	**85.69**	**80.11**
es ancora	85.11	79.52	**85.61**	**80.18**
fr gsd	**84.35**	77.59	84.21	**77.94**
fro	**82.32**	**74.24**	78.91	69.95
gl [600] treegal	**83.80**	**78.06**	83.60	77.63
it isdt	87.03	81.67	**87.10**	**82.27**
la proiel	**66.25**	**58.88**	65.07	57.80
pt bosque	**84.93**	79.37	84.90	**79.83**
ro rrt	79.83	70.46	**79.93**	**70.88**
fa	78.76	72.95	**79.93**	**74.07**
hi hdtb	**89.32**	**82.89**	88.75	82.60
kmr [20]	**69.08**	**59.64**	54.77	45.07
mr	**78.65**	**68.97**	76.60	64.04
ur	84.32	77.02	**84.82**	**78.19**
el	86.44	83.30	**86.88**	**83.96**
grc proiel	**73.82**	**67.88**	71.68	66.05
ga [566]	75.91	67.54	**76.20**	**67.72**
hy [50]	**65.03**	**51.76**	59.27	46.67
id gsd	**81.08**	**74.97**	80.83	74.69
ja gsd	91.22	87.31	**91.40**	**87.37**
ko kaist	73.38	68.35	**74.23**	**69.81**
kk [31]	**70.82**	**55.42**	62.81	44.59
tr imst	**59.64**	**50.66**	59.00	50.54
ug	**66.33**	**48.20**	63.66	46.07
et	**75.32**	**68.13**	73.91	66.96
fi ftb	**78.05**	**72.20**	74.66	68.22
hu	79.51	72.88	**80.15**	**74.31**
sme [2257]	**80.13**	**76.40**	78.34	74.25
ta	75.05	66.94	**76.19**	**67.93**
te	**88.88**	**74.24**	87.01	72.05
vi	65.59	61.15	**66.02**	**61.74**
zh	**80.36**	**74.79**	80.14	74.52
Avg	80.05	73.35	79.47	73.02
Avg No Dev	70.97	60.69	63.93	53.05

Table 1: Parsing results for languages with a training set for phylogenetic models and independent models. The training set size of languages without a developpement set are reported in brackets.

Lang	Model	UAS	LAS
am	Semitic	57.27	26.25
br	Celtic*	61.36	43.89
fo	North-Germanic	52.40	46.52
sa	Indic	56.18	40.46
kpv lattice	Finno-Permiac*	65.16	52.11
pcm	World	60.43	43.80
th	World	29.14	17.61
tl	Austronesian*	70.89	50.38
wbp	World	87.67	65.66
yo	World	56.16	37.51
yue	Sino-Tibetan*	41.68	25.02
Avg		58.04	40.83

Table 2: Accuracy of languages without a training set.

tives. While Buryat (bxr) that has a very small training set benefits from universal linguistic information and gain almost 11 points UAS, Basque (eu) that has a very different grammatical structure than other languages and enough training data (5396 sentences) looses 3.25 LAS. Gains and losses are marginal for the other five languages (id, ja, ko, vi, zh).

Overall results are a bit below the state of the art, but the model is very simple and relies on gold morphology, so it is not really comparable.

3.3 Zero-Shot Parsing

Table 2 reports parsing results for languages that do not have a training set. Because of phylogenetic training and the tree structure that guides it, it can happen that a language ancestor's model is in fact trained on data only accounting for a narrow range of later stages. For example, while Faroese uses the North-Germanic model refined on both Norwegians, Swedish and Danish data, Tagalog uses the Austronesian model only refined with Indonesian data thus making it more an Indonesian model than an actual Austronesian model. Those cases are marked by an asterisk in the table. Komi (kpv) model is refined on Finno-Samic data, Breton (br) model on Irish data, Cantonese (yue) model on Mandarin data.

Looking at Table 2, we make the following observations. As expected scores are on average lower than for languages with training data, however the UAS/LAS gap is substantially bigger from 6.781 to 17.08 points. It is hard to compare to other works on zero-shot parsing since they use different data and scores span a big range, but our results are comparable to those of Aufrant et al. (2016) and Naseem et al. (2012), while our zero-shot models are given for free by the phylogenetic training method.

On a language per language basis, we see that there are a few important factors, the most striking being genre. Tagalog (tl) and more surprisingly Warlpiri (wbp) have relatively high parsing accuracy despite being either completely isolated or having only one relative (Indonesian). This is likely because their data are well annotated stereotypical sentences extracted from grammars, thus making them easy to parse.

Then we see that Naija (pcm) and Yoruba (yo) are about 25 points higher than Thai (th) despite them three having low morphology (in the treebanks). As they have different genres (spoken, bible, news and wiki), without a deeper look at the trees themselves, our best guess is that this is due to Thai having a different script. Naija and Yoruba both use the Latin alphabet, and as such they can rely to some extent on the character model to share information with other languages, to at least organise the character space. This analysis would also carry for Cantonese (yue). It is a morphologically simple language, and despite having a relative (Mandarin), its score is rather low. The genre alone (spoken) would not explain everything as Naija has also a spoken treebank and a higher score. The writing system might be at blame once again. Indeed, Chinese characters are very different from alphabetic characters and are much harder to use in character models because of sparsity. Comparing Mandarin and Cantonese test sets with Mandarin train set, the amount of out-of-vocabulary words is 32.47% of types (11.90% of tokens) for Mandarin and 54.88% of types (56.50% of tokens) for Cantonese. The results for out-of-vocabulary characters are even more striking with 3.73% of types (0.49% of tokens) for Mandarin and 12.97% of types (34.29% of tokens) for Cantonese. This shows that not only there are a lot of OOV in Cantonese test set, but that those words/characters are common ones as 12.97% of character types missing make up for more than a third of all character tokens missing, where on the contrary Mandarin OOV are seldom and account for less tokens percentage than types. This is one more argument supporting the importance of the character vector.

Other important factors are typology and morphology. Amharic (am) despite its unique script has a higher score than Cantonese that actually shares its scripts (to some extent as we have seen) with Mandarin. The key point for Amharic score, is that all its relatives (Hebrew, Arabic and Coptic) have their own scripts and are morphologically rich, thus the model learns to use morphological information. The analysis is similar for Komi which on top of sharing morphology with its relatives also share the writing system which provides it an extra gain. However, this might word in the opposite direction as well, as we can see with Faroese, Breton and Sanskrit. Faroese (fo) is morphologically rich and that should help, however its North-Germanic relatives are morphologically much simpler. Thus the model does not learn to rely on morphological attributes nor on word endings for the character model as much. The same is true for Sanskrit (sa), which is morphologically richer than its modern Indic relatives, with an extra layer of specific writing systems. Eventually, Breton model (br) is refined over Irish data only and while Irish is a typological outlier amongst Indo-European languages because of its Verb-Subject-Object word order, Breton has the standard Subject-Verb-Object, thus using Irish data might actually be detrimental.

These arguments show the respective importance of the writing system, the genre of the data, the morphological analysis and the typology in phylogenetic zero-shot dependency parsing. Those factors can either work together positively (Komi) or negatively (Cantonese) or cancel each other out (Amharic, Faroese).

4 Related Work

The goal of multi-task learning is to learn related tasks (either sharing their input and/or output space of participating of the same pipeline) jointly in order to improve their models over independently learned one (Caruana, 1997). In Søgaard et al. (2016), task hierarchy is directly encoded in the neural model allowing tasks with different output space to share parts of their parameters (POS tagging comes at a lower level than CCG parsing and only back-propagates to lower layers). Likewise, in Johnson et al. (2017), the encoder/decoder architecture allows to learn encoders that target several output languages and decoders than handle data from various input languages. However, in multi-task learning literature, task relationships are often fixed. In Cavallanti et al. (2010) tasks with the same output spaces share parameter updates through a fixed similarity graph. In this work, changing level in the tree can be seen as splitting the similarity graph into disjoint sub graphs. It is

a way to have tasks relationships evolving during training and to encode information about task evolution that lacks in other multi-task methods.

In multi-lingual parsing, Ammar et al. (2016) propose to train a single model to parse many languages using both typological information, cross-lingual word representations and language specific information. While their model gives good results, they only apply it to 7 Germanic and Romance languages. It would be worth doing the experiment with 50+ languages and see how the results would change. However, because of language specific information their model would probably become very big. In this work, language specific information is not added on the top of the model, but is just language generic information that refines over time.

Che et al. (2017; 2018) and Stymne et al. (2018) propose to train parsers on several concatenated treebanks either from the same language or from related languages and to fine-tune the parsers on individual treebanks afterward to fit specific languages/domains. The main difference with our method, is that instead of one step of fine-tuning, we perform as many fine-tuning as there are ancestors in the tree, each time targeting more and more specific data. This in turn requires that we handle data imbalance therefore using sampling rather than plain concatenation.

Aufrant et al. (2016) propose to tackle zero-shot parsing by rewriting source treebanks to better fit target language typology. Assuming that typology is homogeneous in a language family, the phylogeny should drive models to be typologically aware. However, as we have seen for Breton and Irish, that assumption might not always hold.

Eventually, the closest work from our in spirit is the one of Berg-Kirkpatrick et al. (2010). They use a phylogenetic tree to guide the training of unsupervised dependency parsing models of several languages, using ancestor models to tie descendent ones. The main difference here beside supervision, is that we do not use ancestor models as biases but rather as initialization of descendent models.

5 Conclusion

We have presented a multi-task learning framework that allows one to train models for several tasks that have diverged over time. Leveraging their common evolutionary history through a phylogenetic tree, models share parameters and training samples until they need to diverge. As a by product of this phylogenetic training, we are provided with intermediary models that can be used to zero-shot a new related task, given its position in the evolutionary history.

We have applied this framework to dependency parsing using a graph-based neural parser and the phylogenetic tree of the languages from UD 2.2 to guide the training process. Our results show that phylogenetic training is beneficial for well populated families such as Indo-European and Uralic. It also helps generalization and prevents over-fitting when very few data are available. For zero-shot parsing, genre, writing system and morphology are crucial factors for the quality of parse predictions.

Some works have been done on automatically learning task relationship in multi-task setting. It would be interesting to see how the algorithm could figure out when and how to cluster languages automatically as phylogenetic trees do not directly depict grammar evolution.

Our model does not know that Latin came before Old French and before modern French, or that despite being Germanic, English underwent a heavy Romance influence. It would be worth investigating softening the tree constraints and instigating more evolutionary information in the structure.

Another important point is that we use gold part-of-speech and morphological information which is unlikely to be available in real scenarios. However, our new training procedure can be applied to any task, so a future work would be to use it to perform phylogenetic POS tagging.

Other directions for the future are designing better sampling methods as well as better ways to measure training convergence at each level.

Acknowledgement

This work was supported by ANR Grant GRASP No. ANR-16-CE33-0011-01 and Grant from CPER Nord-Pas de Calais/FEDER DATA Advanced data science and technologies 2015-2020. We also thank the reviewers for their valuable feedback.

References

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

Lauriane Aufrant, Guillaume Wisniewski, and François Yvon. 2016. Zero-resource Dependency Parsing: Boosting Delexicalized Cross-lingual Transfer with Linguistic Knowledge. In *COLING 2016, the 26th International Conference on Computational Linguistics*, pages 119–130, Osaka, Japan. The COLING 2016 Organizing Committee.

Taylor Berg-Kirkpatrick and Dan Klein. 2010. Phylogenetic grammar induction. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 1288–1297.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

Giovanni Cavallanti, Nicolò Cesa-Bianchi, and Claudio Gentile. 2010. Linear algorithms for online multitask classification. *J. Mach. Learn. Res.*, 11:2901–2934.

Wanxiang Che, Jiang Guo, Yuxuan Wang, Bo Zheng, Huaipeng Zhao, Yang Liu, Dechuan Teng, and Ting Liu. 2017. The hit-scir system for end-to-end parsing of universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 52–62, Vancouver, Canada. Association for Computational Linguistics.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better ud parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64. Association for Computational Linguistics.

Xinying Chen and Kim Gerdes. 2017. Classifying languages by dependency structure. typologies of delexicalized universal dependency treebanks. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017), September 18-20, 2017, Università di Pisa, Italy*, 139, pages 54–63. Linköping University Electronic Press, Linköpings universitet.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the conll 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

Jason M. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, pages 340–345, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Gerhard Jäger. 2015. Support for linguistic macrofamilies from weighted sequence alignment. *Proceedings of the National Academy of Sciences*, 112(41):12752–12757.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 629–637, Stroudsburg, PA, USA. Association for Computational Linguistics.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, Paul Michel, Yusuke Oda, Matthew Richardson, Naomi Saphra, Swabha Swayamdipta, and Pengcheng Yin. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, John Bauer, Sandra Bellato, Kepa Bengoetxea, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Rogier Blokland, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas,

Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Radu Ion, Elena Irimia, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Tolga Kayadelen, Václava Kettnerová, Jesse Kirchner, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Shinsuke Mori, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Adédayọ̀ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roșca, Olga Rudina, Shoval Sadde, Shadi Saleh, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Yuta Takahashi, Takaaki Tanaka, Isabelle Tellier, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Veronika Vincze, Lars Wallin, Jonathan North Washington, Seyi Williams, Mats Wirén, Tsegay Woldemariam, Tak-sum Wong, Chunxiao Yan, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. 2018. Universal dependencies 2.2. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Natalie Schluter and Željko Agić. 2017. Empirically sampling universal dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 117–122.

Gary F. Simons and Charles D. Fennig, editors. 2018. *Ethnologue: Languages of the World, Twenty-first edition.* SIL International, Dallas, TX, USA.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*.

Sara Stymne, Miryam de Lhoneux, Aaron Smith, and Joakim Nivre. 2018. Parser training with heterogeneous treebanks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 619–625, Melbourne, Australia. Association for Computational Linguistics.

A Appendix

	Phylogenetic		Independent	
	UAS	LAS	UAS	LAS
Afro-Asiatic	80.73	74.99	**80.90**	**75.60**
Indo-European	**80.41**	**73.73**	78.93	72.45
Germanic	80.64	74.58	**80.82**	**75.16**
Slavic	**80.83**	**74.37**	78.21	72.09
Romance	**82.03**	**75.40**	81.67	75.18
Indo-Iranian	**80.03**	**72.29**	76.97	68.79
Greek	**80.13**	**75.59**	79.28	75.01
Turkic	**65.60**	**51.43**	61.82	47.07
Uralic	**78.25**	**72.40**	76.76	70.93
Dravidian	**81.97**	**70.59**	81.60	69.99
Avg	80.05	73.35	79.47	73.02
Avg No Dev	70.97	60.69	63.93	53.05

Table 3: Parsing results for phylogenetic and independent neural models averaged by language family. Families are sorted in the same order as they appear in Table 1. Indo-European averages include Armenian (hy) and Irish (ga). Global averages are repeated for completeness. Best results are reported in bold.

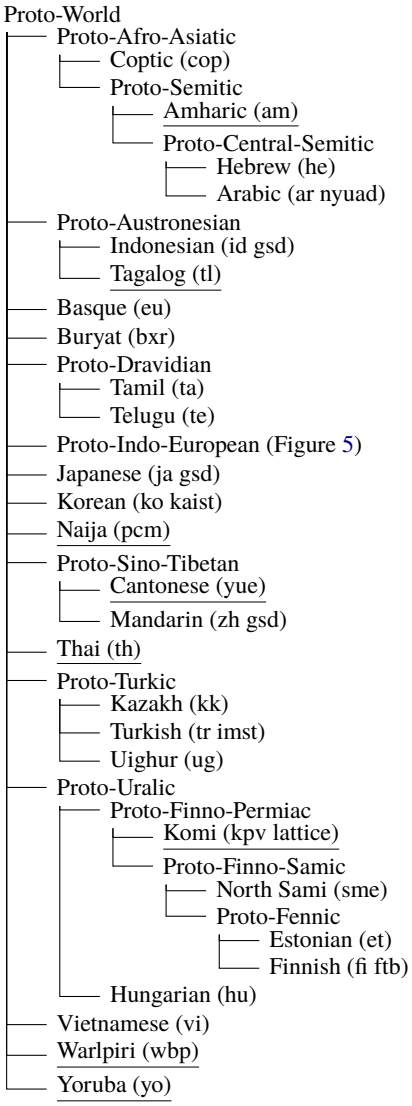

Figure 4: Phylogenetic tree used to guide the training process of the multi-lingual parser. Underlined languages are those that do not have a training set. The code of the language and if necessary the name of the treebank are given in parentheses. The Indo-European sub-tree is depicted on the right.

Figures 4 and 5 represent the phylogenetic tree used for guiding the training process. As we only use data from the UD project 2.2, we collapse unique child so that Vietnamese is not an Austro-Asiatic language, it is just Vietnamese. We also only use well attested families, thus Buryat, a Mongolic language, is alone and not linked to Turkic languages. Maybe, the most disputable choice is to put Naija in its own Creole family instead of the Germanic family.

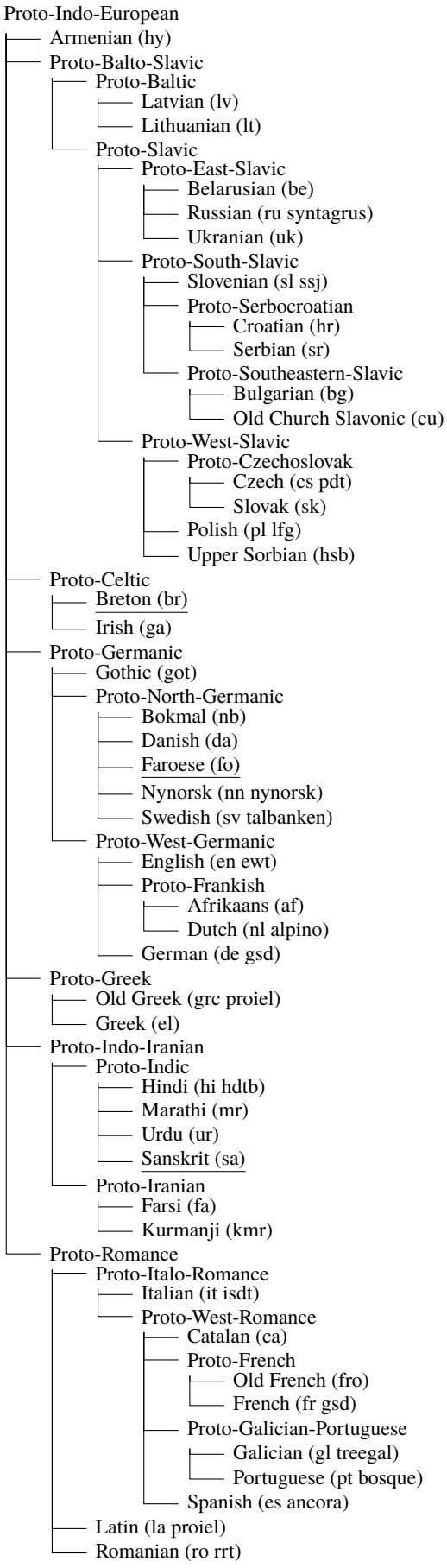

Figure 5: The Indo-European phylogenetic tree.