A Character-Based Intersection Graph Approach to Linguistic Phylogeny

Jessica Enright
University of Alberta
Edmonton, Alberta, Canada
enright@cs.ualberta.ca

Abstract

Linguists use phylogenetic methods to build evolutionary trees of languages given lexical, phonological, and morphological data. Perfect phylogeny is too restrictive to explain most data sets. Conservative Dollo phylogeny is more permissive, and has been used in biological applications. We propose the use of conservative Dollo phylogeny as an alternative or complementary approach for linguistic phylogenetics. We test this approach on an Indo-European dataset.

1 Introduction

1.1 Language Phylogeny

A linguistic phylogenetic tree is a tree describing the evolution of some set of languages. Usually, we build such a tree using information given by a set of characters associated with those languages.

We say that a character *back-mutated* if after evolving from 0 state to 1 state, it subsequently is lost and switches back on the tree from 1 state to 0 state. We say that a character has *parallel evolution* if it evolves twice on the tree from state 0 to state 1 independently. We say that a character is *borrowed* if, on the true evolutionary tree, it has been transfered from one branch to another by contact between linguistic groups. Loanwords are an example of this.

1.2 Perfect phylogeny

Given a set of binary characters $\mathcal{C} = \{c_1...c_j\}$, we say that a rooted tree $T = (r, V_T, E_T)$ with languages $\mathcal{L} = l_1...l_k$ as the leaf nodes of T is a *perfect phylogeny* if there is a binary labeling of each character at each node such that the root node is labeled with a zero for each character, and for each character both the subtree induced by the nodes labeled 1 at that character, and the subtree induced by the nodes labeled 0 at that character are connected.

This means that each character evolves exactly once, and that there is no back-mutation or borrowing.

We can recognize whether a set of characters admits a perfect phylogeny in polynomial time (Felsenstein, 2004). Unfortunately, often character data does not admit a perfect phylogeny.

Usually the question given character data is: How far away is this data from admitting a perfect phylogeny? What is the minimum level of borrowing, back mutation or parallel evolution that we must allow to produce a tree that describes this data? Answering this question is NP-Hard (Day et al., 1986).

Many approaches describe and formalize this question. Nakhleh et al. (2005b) provide an excellent survey of linguistic phylogenetic methods.

Nakhleh et al. (2005a) proposed perfect phylogeny networks as a way of considering the phylogeny problem. A perfect phylogeny network is a graph that is not required to be a tree such that every character exhibits a perfect phylogeny on at least one of the subtrees of that graph.

Unfortunately, even given a phylogenetic tree and character data, determining the minimum number of edges one must add to produce a perfect phylogeny network is NP-Hard (Day et al., 1986). Nakhleh et al. (2005a) mention that applying the perfect phylogeny network approach to their Indo-European language dataset is tractable only because one need only add very few edges to their tree to produce a perfect phylogeny network.

1.3 Dollo Phylogenies

In contrast to a perfect phylogeny, a Dollo phylogeny allows an arbitrary number of back mutations.

Given a set of binary characters $\mathcal{C} = \{c_1...c_j\}$, we say that a rooted tree $T = (r, V_T, E_T)$ with

Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing, ACL 2010, pages 42–46,
Uppsala, Sweden, 16 July 2010.

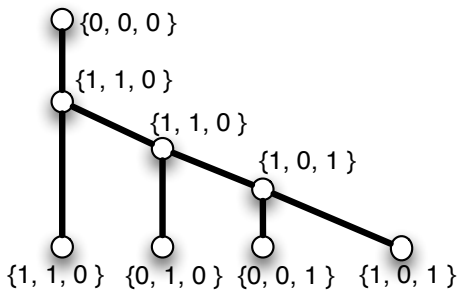

Figure 1: A tree that is a conservative Dollo phylogeny, but not a perfect phylogeny, as characters one and two back-mutate. The lists at each node contain the state of characters one, two, and three in that order.

languages $\mathcal{L} = l_1...l_k$ as the leaf nodes of T is a *Dollo phylogeny* if there is a binary labeling of each character at each node such that the root node is labeled with a zero for each character, and for each character the subtree induced by the nodes labeled 1 is connected.

This means that each character evolves exactly once but an arbitrary number of back-mutations are allowed. Unfortunately, every set of character data admits a Dollo phylogeny. Clearly Dollo phylogeny is too permissive to be a useful notion in linguistic phylogenetics.

Przytycka et al. (2006) discussed the idea of a *conservative Dollo phylogeny*.

Given a set of binary characters $\mathcal{C} = \{c_1...c_j\}$, we say that a rooted tree $T = (r, V_T, E_T)$ with languages $\mathcal{L} = l_1...l_k$ as the leaf nodes of T is a *conservative Dollo phylogeny* (CDP) if there is a binary labeling of each character at each node such that the root node is labeled with a zero for each character, for each character the subtree induced by the nodes labeled 1 is connected, and if two characters appear together in their 1 states in the tree at an internal node, they also occur together in their 1 states in the tree at a leaf node. Recall that the leaves in this tree are the languages for which we have data. For an example, see Figure 1.

If two characters existed together in some ancestral language, they must also exist together in at least one leaf language. That is, if they have ever existed together in the same language, we have evidence of it in the form of a known language that possessed both of those characters. Is this a reasonable assumption? We have no evidence that it is. However, it's certainly a more reasonable assumption than that required for a perfect phylogeny. We expect that often, data sets will not admit a CDP, and that, like for perfect phylogeny, the question will be: How far away are the data from admitting a CDP?

Przytycka et al. (2006) prove that a set of characters admit a CDP if and only if their intersection graph is chordal. Chordal graphs are graphs with no induced cycles longer than three vertices. Rose et al. (1976) provide a linear-time recognition algorithm for chordal graphs.

Graph $G = (V, E)$ is an intersection graph of a family of sets $\mathcal{S}$ if there is a bijection $\mathcal{F}$ between V and $\mathcal{S}$ such that for every two sets $s, t \in \mathcal{S}$ $\mathcal{F}(s)$ is adjacent to $\mathcal{F}(t)$ if and only if s intersects t. Set s intersects set t if they share at least one element. Given sets, we can compute their intersection graph in linear time. For an example of an intersection graph derived from a family of sets, see Figure 2.

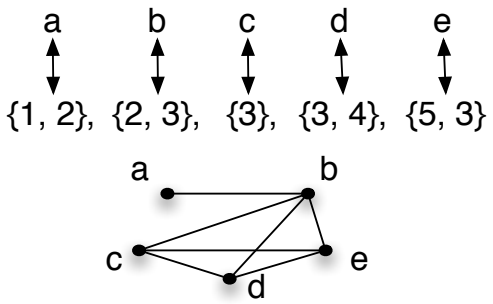

Figure 2: An example of a family of sets labeled a, b, c, d, e on the top and the intersection graph of those sets on the bottom.

We can then determine if a set of characters admits a CDP in linear time. This approach to phylogeny was used by Przytycka et al. (2006) in a biological phylogenetic application. Here, we use it for linguistic phylogeny.

2 Methodology

We implemented an algorithm to, given a character dataset, compute the intersection graph of those characters, and determine whether the resulting graph is chordal as given by Rose et al. (1976). This tells us whether or not the dataset admits a CDP. We also implemented an exhaustive search that computes the minimum number of characters that must be borrowed to otherwise admit a CDP.

We ran our program on the Indo-European character dataset used by Nakhleh et al. (2005a), and available online at http://www.cs.rice.edu/ nakhleh/CPHL/.

2.1 Language Family Grouping

Nakhleh et al. (2005a) combined established language groups into a single language during computation to decrease computation time. We use the same families as they do, and do the same in two of our experiments.

For example, we consider the Tocharian language family, consisting of Tocharian A and Tocharian B to be a single language when building our intersection graph. This language grouping is done as a preprocessing step to the construction of the intersection graph of the characters.

We expect this transformation to be particularly useful in the CDP setting, beyond just decreasing computation time. We expect it will make our data closer to admitting a CDP in a way consistent with true evolutionary history.

Consider the difference between the intersection graph of a set of characters with family grouping and without. Let s and t be two characters that, are considered to intersect with family grouping, but not without. Then s and t are not present in any of the same languages, but there are two languages l_i, l_j such that l_i has character s but not t and language l_j has character t but not s, and l_i and l_j are in the same family L.

We use the language family definitions given by Nakhleh et al. (2005a), where these language families are identified as consistent with all characters, and it is argued that it is very unlikely there is any borrowing between a portion of the tree inside the family, and a portion of the tree outside the family.

Therefore, if s and t are both present within leaves in the language family L, and neither is borrowed from outside the family, then each of s, t is either present only within language family L, or is present in at least one internal node ancestral to language family L. If s and t are only present within the language family, they are not informative when language family grouping is used.

However, if both s and t are present at an internal node ancestral to language family L, then this is important information that we have derived by applying family language grouping, and will make the data closer to admitting a CDP in terms of number of borrowings required.

2.2 Binary Data

We made the data binary by separating states of a given character as best indicated by notes provided by Nakhleh et al. (2005a) on their coding of the characters. In making the data binary, we have likely lost some constraining information. When a language (or language family, when that grouping was used) has a unique state at a character, we coded this as having all possible non-ancestral states. The basis for this is that some of these codes indicate that there is no data for that character at that language, or that if that language actually does have a unique state at that character, it is uninformative, but could have evolved from any other state. Data processing by someone more highly trained in linguistics would either confirm this decision or provide an alternative approach. We have tried to remain as close as possible to how the data is used in Nakhleh et al. (2005a).

3 Experiments

We ran four experiments to investigate the usefulness of the conservative Dollo parsimony approach. We ran our implementation on:

1. All characters without language family grouping
2. All characters with language family grouping
3. Phonological and morphological characters only without language family grouping
4. Phonological and morphological characters only with language family grouping

4 Results

We give our results in Table 4

For the morphological and phonological dataset, both grouped and ungrouped, we extracted a phylogenetic tree from our program's output. These trees were consistent with Tree A in (Nakhleh et al., 2005a). The fact that we managed to build a tree consistent with expectations without any input tree is very encouraging.

Recall that when we use language grouping we combine all languages identified as being from an established family by Nakhleh et al. (2005a) into a single language. For example, instead of considering both Tocharian A and Tocharian B, in our experiments with language grouping we consider a single language, Tocharian, that has all characteristics of Tocharian A and all characteristics of Tocharian B.

Table 1: The results of conservative Dollo phylogeny checking algorithm on modified versions of the Indo-European character dataset as used in (Nakhleh et al., 2005a). We ran each program for at most 1 hour. Entries of "Too slow" indicate that we did not allow the program to halt.

Dataset	Admits a CDP?		Minimum number of languages that must borrow	
	Answer	Time	Answer	Time
Phonological, Morphological Data without Language Grouping	Yes	<1 s	0	<1 s
Phonological, Morphological Data with Language Grouping	Yes	<1 s	0	<1 s
All Data without Language Grouping	No	<1 s	-	Too slow
All Data with Language Grouping	No	<1 s	2	< 1 s

In our experiments without language grouping, we do not combine languages in this way, and instead consider all 24 languages separately.

5 Discussion

When is the CDP approach useful for linguistic phylogenetics?

Because a CDP allows back-mutation, it is likely most useful for datasets that exhibit a lot of back mutation, and not a lot of borrowing. Phonological and morphological characters are more likely to fit this requirement than lexical data. This is reflected in our positive results on the phonological and morphological characters alone.

In contrast, when we included the lexical data, the dataset did not admit a conservative Dollo parsimony, whether or not we used language family grouping. We expect this is due to borrowing of lexical characters.

The full dataset with language family grouping was much closer to admitting a conservative Dollo parsimony than the full dataset without language family grouping. As explained in our Methodology section, this was expected and reinforces our position that language family grouping is extremely useful when computing conservative Dollo phylogenies.

Our experiments ran in either negligible time, or were not allowed to halt. The speed of the fast experiments suggests that computing conservative Dollo phylogenies might be useful in constructing a tree when no tree is known, and the amount of character data causes computing other types of phylogenies to be intractable.

6 Future Work

We are currently pursuing several extensions to this work.

First, we are developing an improved heuristic search for the minimum number of edges that need to be removed from or added to a graph to make the resulting graph chordal. This will enable us to use the Dollo phylogeny approach outlined here on character data sets that require more borrowing to fully explain them.

Using this improved search, we will run experiments on other sets of character data.

Nakhleh et al. (2005a) started with several proposed trees in their work on perfect phylogenetic networks. We plan to implement a version of our CDP approach that takes as input a proposed tree. This version will calculate the minimum number of edges that must be added to create a Dollo phylogeny network, as analogous to Nakhleh et al.'s perfect phylogenetic network. This minimum number of edges would be useful as a lower bound for the required number of edges one must add to produce a perfect phylogeny network.

7 Conclusion

We have presented an alternative phylogeny that may be of use in linguistic phylogenetics, particularly on phonological or morphological data. We have proposed a number of future extensions based on our experiments that we hope will improve the performance of this approach.

Acknowledgments

The author would like to acknowledge the helpful input of reviewers, as well as Dr. Gzegorz Kondrak and Dr. Lorna Stewart.

References

William Day, David Johnson, and David Sankoff. 1986. The computational complexity of inferring rooted phylogenies by parsimony. *Mathematical Biosciences*, 81:33–42.

Joseph Felsenstein. 2004. *Inferring Phylogenies*. Number 1. Sinauer Associates, Massachusetts, USA.

Luay Nakhleh, Don Ringe, and Tandy Warnow. 2005a. Perfect phylogenetic networks: A new methodology for reconstructing the evolutionary history of natural languages. *Language (Journal of the Linguistic Society of America)*, 81(2):382–420.

Luay Nakhleh, Tandy Warnow, Don Ringe, and Steven N. Evans. 2005b. A comparison of phylogenetic reconstruction methods on an ie dataset. *The Transactions of the Philological Society*, 3(2):171 – 192.

Teresa Przytycka, George Davis, Nan Song, and Dannie Durand. 2006. Graph theoretical insights into evolution of multidomain proteins. *Journal of computational biology : a journal of computational molecular cell biology*, 13(2):351–363.

Donald J. Rose, R. Endre Tarjan, and George S. Leuker. 1976. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal of Computing*, 5(2):266–283.