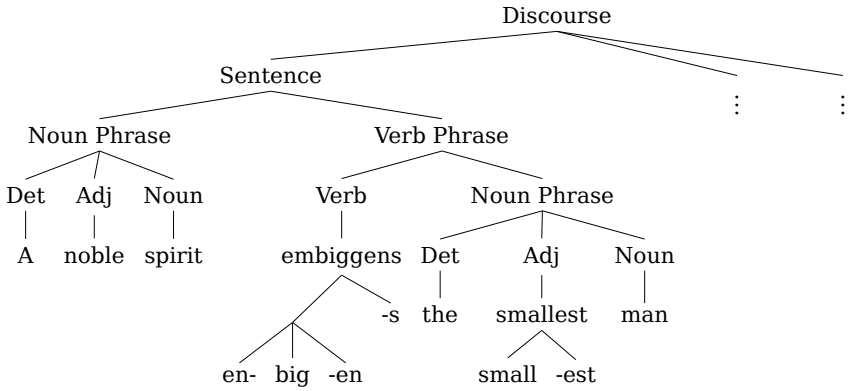
What can computational linguistics do for computational stemmatology?

David Chiang
Information Sciences Institute
University of Southern California
chiang@isi.edu

28 January 2010

Computational linguistics aims to develop computational models of human language. Research in computational linguistics ranges from the theoretical (e.g., asking what kind of formal automaton the human language faculty might be) to the practical (e.g., building systems to translate massive collections of text), but across the board, two recurring themes of computational linguistics are *structure* and *ambiguity*.

The *structure* of language is hierarchical. Though it appears on the surface to be a stream of symbols, a writing or conversation is organized into sentences, the sentences into phrases, the phrases into smaller phrases, the phrases into words, and the words into morphemes:



A complete account of language would have to take all these levels of structure into consideration. Because this structure is not always overt, we are constantly faced with the possibility of *ambiguity*, that is, a text having more than one possible structure. And because structure is hierarchical, the number of possible structures we have to work with is typically exponential.

I think there are two broad areas where computational linguistics can be helpful for computational stemmatology. First, although computational stemmatology has drawn on ideas from computational biology with great success, based on an analogy between genes and language, nevertheless, in order to model transmission of

natural-language texts one must be attentive to the structure of natural language. Computational linguistics may provide useful insights into improving models of textual change. Second, natural-language-processing algorithms routinely deal with exponentially sized spaces of possible tree structures. Many tricks of the computational linguistics trade are focused on managing this complexity. These techniques may also prove useful for searching through the space of tree topologies.

1 Modeling textual change

A common assumption in reconstruction of evolutionary trees is to partition a sequence into segments and assume that the segments change independently of one another. For example, a line from Chaucer might be partitioned as:

In	pacience	she ladde	a symple	lyf
In	pacience	ladde	a ful symple	lyf
In	pacience	ladde	a symple	lyf
In	pacience	ladde she	a ful symple	lyf
In	pacience	hadde	a ful symple	lyf
In	pacience	ladde she	hur	lyf
In	pacience	ladde	a ful mery	lyf

and the space of all possible hypothesized texts is formed by taking one chunk from each column (in computational linguistics, this is known as a *sausage lattice*).

But this partitioning somewhat lacks flexibility. For example, the change `ladde` ↔ `hadde` only involves a single letter, `ladde` ↔ `she ladde` a single word, and `she ladde` ↔ `ladde she`, multiple words. We would like the model to handle each in an appropriate manner, and to make appropriate generalizations from each. By working at the grapheme level, the model can learn whether `l` ↔ `h` is a likely or unlikely change; by working at the word level, it can learn whether insertion/deletion of pronouns is likely or unlikely; and by working at the syntactic level, it can learn whether inversion of subjects and verbs is likely or unlikely.

All this requires careful modeling of linguistic structure, which computational linguistics can provide. In particular, the subfield of *machine translation* specializes in modeling transformations of natural language text, and recent progress in this area has yielded models which are ever more sensitive to linguistic structure [1, 4]. Though these models were designed for transformations across languages, they can also be applied to transformations within a single language.

As an aside, the use of machine-translation models suggests the more ambitious goal of incorporating evidence from early translations of a text. The utility of such evidence seems to be an open question, and it would be significant if computer models could help integrate this evidence more effectively.

2 Searching the space of trees

The biggest algorithmic challenge in phylogeny reconstruction is dealing with the space of all possible tree topologies: searching for the optimal tree, or, in some learn-

ing algorithms, calculating the expected value of some variable over all possible trees.

If we keep the topology of the tree fixed, it is easy to search or calculate expectations over all the interior node labels (the unobserved ancestor texts). For example, the classic model of Felsenstein [2] uses Expectation-Maximization (EM) to learn a model in such a setting. Conversely, it turns out that if we keep the number of nodes and their labels fixed, it is also easy to search for an optimal tree topology, using the Chu-Liu-Edmonds algorithm [9] to find a maximum spanning tree on a directed graph, or to calculate expectations, using Kirchhoff's Matrix-Tree Theorem [10]. This insight has recently been applied to *nonprojective dependency parsing*, which is a search for an optimal tree whose nodes are the words of a sentence, in any order [6, 8, 5].

But the real algorithmic challenge is simultaneously considering all possible trees and all possible labels for the interior nodes. In dependency parsing, we encounter this problem when, for example, doing simultaneous parsing and part-of-speech tagging. We think an extension of Friedman et al.'s approximate EM algorithm [3] along the lines of Smith and Eisner's approach to dependency parsing using loopy belief propagation [7] may provide a good solution to this problem.

References

[1] David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proc. 43rd Annual Meeting of the ACL*, pages 263–270, 2005.

[2] Joseph Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *J. Molecular Evolution*, 17:368–376, 1981.

[3] Nir Friedman, Matan Ninio, Itsik Pe'er, and Tal Pupko. A structural EM algorithm for phylogenetic inference. *J. Computational Biology*, 9:331–353, 2002.

[4] Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. Scalable inference and training of context-rich syntactic translation models. In *Proc. COLING-ACL 2006*, pages 961–968, 2006.

[5] Terry Koo, Amir Globerson, Xavier Carreras, and Michael Collins. Structured prediction models via the Matrix-Tree Theorem. In *Proc. 2007 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2007.

[6] Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. Non-projective dependency parsing using spanning tree algorithms. In *Proc. HLT/EMNLP 2005*, 2005.

[7] David A. Smith and Jason Eisner. Dependency parsing by belief propagation. In *Proc. 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 145–156, 2008.

[8] David A. Smith and Noah A. Smith. Probabilistic models of nonprojective dependency trees. In *Proc. 2007 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2007.

[9] R. E. Tarjan. Finding optimum branchings. *Networks*, 7:25–35, 1977.

[10] W. T. Tutte. *Graph Theory*. Cambridge UP, 2001.