Research Statement — Jason Eisner

With the advent of ubiquitous speech recognition and massive bodies of informative text on the Web, we will increasingly expect computers to make sense of the kind of language we use naturally.

Computational linguistics is the study of this rich and intricate domain. It is concerned with (1) the algebraic structure of human language; (2) the deep statistical properties of language (for inferring the iceberg from its tip); (3) the systematic relations among sentences, logical formulae, and sound waves; and (4) efficient algorithms for producing, interpreting, and learning language. Recently it has also become a testbed and source of challenges for machine learning.

My research develops robust computational methods for some of the central problems involving automatic language processing. I have often cast these as function optimization problems:

- Find the most probable analysis of a written sentence.
- Find the most plausible grammar for a language.
- Find the best way to pronounce a word in context.
- Find the Web page of greatest interest to a user.

In attacking these problems, I have tried to identify the most useful **linguistic** insights and recast them as principled **mathematics**. On one hand, success depends on choosing the right function to optimize: What does it *mean* for a system of grammatical rules to be a plausible explanation of the observed data? On the other hand, the function must be formally clean and simple enough that one can develop tractable *algorithms* for optimizing it.

Basic Research

Statistical Parsing — *"Find the most probable analysis of a written sentence."*

Accurate parsing of natural language—discovering the recursive structure of an arbitrary sentence—is a major roadblock for systems that attempt to understand or translate text. A great many word meanings and grammatical constructions are *possible* (if rare) in the language. The typical sentence therefore has combinatorially many parses. The task is to output the *right* parse.

The solution is to search not for *possible* parses but for *probable* ones. One may recursively define a probability for every parse tree, such that unusual configurations in a parse tree lower its probability. Finding the most probable parse is now a matter of dynamic programming. Moreover, low-probability parts of the search space can be pruned away for efficiency.

But what kind of probability model best captures linguistic preferences? While the general rule is to estimate the model's parameter values automatically, using training data, what features of parse trees should those parameters pick up on, and how?

- One crucial linguistic ingredient turns out to be rough plausibility of meaning: The verb "solve" not only needs a direct object, but strongly prefers direct objects like "problem" and "puzzle." With Mark Jones, in 1992, I built the first probabilistic parser to include parameters for such word-to-word relationships. Adding these parameters enabled that early parser to identify the correct parse for 96%, rather than 79%, of those test sentences in our (restricted) domain for which it found any parse at all.

- The mathematical structure of the model is also important. In 1994, published 1996, I built one of the first successful real-world parsers, which achieved a state-of-the-art 93% accuracy on real newspaper text. I used the parser to develop and compare several contrasting probability models: these deployed the same linguistic features in different ways. The more elegant models produced significantly more accurate parses.

I also devised a novel dynamic programming parsing algorithm, which can consider all the crucial word-to-word ("bilexical") relationships while preserving worst-case time $O(n^3)$. The "obvious" solution—used by at least three other recent parsers—takes time $\Theta(n^5)$.

Grammar induction (thesis) *"Find the most plausible grammar for a language."*

Current statistical parsers *(see above)* must be trained on sample parses—which are expensive to produce by hand. The parser builds complete parse trees out of small, probable chunks of tree. It must know what chunks are available in the language: an approximate list can be extracted from sample parses. Unfortunately, since available training data are limited, such a list typically omits 5–20% of the chunks needed to parse a new sentence correctly.[1]

What we need is the ability to **generalize statistically** to new chunks from the chunks observed in training data. Suppose training data indicate only that "open" allows a direct object ("open the door"). We might guess that "open" also has a corresponding passive form ("the door was opened"), since those two *kinds* of chunks are correlated for other verbs.

Such generalizations certainly exist in any language. Linguists call them *transformations*. But linguists have no automatic procedure for discovering transformations or exploiting them in a probabilistic context.

My thesis presents a statistical technique for finding such generalizations using the existing data. The approach is declarative. First we quantify a notion of a good grammar (where a grammar is a probability distribution over all tree chunks). Then we search for the *best* grammar, using gradient ascent in the continuous space of all possible grammars.

A good grammar both *describes* the training data and also *generalizes* from it. Bayesian statistics combines these desiderata in a principled way. The most probably correct grammar is, by definition, the one that maximizes Pr(grammar | training data). By Bayes' Theorem, this probability is proportional to

$$\Pr(\text{training data} \mid \text{grammar}) \cdot \Pr(\text{grammar})$$

The first factor emphasizes description, the second generalization. The first factor is high if the grammar assigns high probability to the chunks of tree that actually appear in the sample parses. The second factor, trickier to define and compute, is the *a priori* probability of the grammar itself—a kind of MDL or model complexity penalization term.

We define this Pr(grammar) factor to be high if the grammar has strong, consistent internal structure. How? I use a kind of Bayesian network (graphical model) for continuous variables, to specify a joint prior distribution over the chunk probabilities. We want transformations to predict most chunks from other chunks. So Pr(grammar) is high if most of the chunk probabilities in the grammar can be approximately predicted from the others, by assuming some fixed set of transformational operators that convert various kinds of chunks into others at particular rates. (A separate joint prior on these rates prevents overfitting and generalizes across similar transformations.)

In a direct comparison, this procedure induces better grammars from English newspaper text than any described in the literature. (The objective measure is cross-entropy, i.e., the ability of a grammar induced from training data to predict unseen test data.) Moreover, the transformational operators it identifies as having high rates in English can be inspected directly and are linguistically plausible. The approach is flexible enough to accomodate a wide range of linguistic frameworks, i.e., notions of what "chunks" and "transformations" should look like in natural language grammars.

[1] These figures are for English *Wall Street Journal* text and a particular notion of chunk, using 50,000 example parses created by a $250,000 effort involving 5 person-years of work. Most languages have far less data available.

Automata Theory and Phonology *"Find the best way to pronounce a word."*

Phonology is a major branch of linguistics. It studies how pieces of words are pronounced in context. Compare *resign-ation* to its stem, *resign*: the "g" is no longer silent, and the vowels and accent have changed systematically too. Every language has its own often bizarre system for contextual pronunciation. A human—or a computer—must know that system in order to speak or comprehend the language properly.

Over the past five years, phonologists have suddenly and wholeheartedly embraced a discrete optimization approach (Optimality Theory). On this view, a language's contextual pronunciations result from its attempt to enforce some set of 100+ formal properties. The correct contextual pronunciation is the one that satisfies as many as possible of the language's most important properties (in a sense that phonologists make precise).

This paradigm shift left phonologists to posit whatever properties they found convenient for each language—which, as they realized, meant they no longer had a unifying, falsifiable theory of possible human pronunciation systems. I have been able to show that this zoo of ad hoc properties can be replaced in practice with formal properties limited to a surprisingly clean and simple mathematical class. (A baby learning a language would be able to consider just that class of properties.)

Under this empirically motivated formalization, I show that

- the difficult discrete optimization problem can now be solved by combining well-understood weighted finite-state methods. Interestingly, a converse reduction also holds.
- optimization is NP-hard on the number of properties (thus the finite-state automata can be exponentially large in the worst case).
- the common case has a fast implementation using a novel, "factored" representation of large automata as intersections of smaller automata.
- these simple properties, unlike some proposed by linguists, cannot be combined to "accidentally" generate certain unwanted phenomena that are not observed in human pronunciation.

Applied Research

Data Mining / Information Retrieval *"Find Web pages of interest to a user."*

The Internet often leaves both users and content providers drowning in a flood of information about each other. A **recommendation system** is a method for automatically identifying objects that are likely to be of interest to a given user. Users can rely on recommendation systems to pick out Web pages and links, news stories, videos, or doctors for them, or to rerank the output of a search engine. Merchants can use such a system to select personalized advertising of interest to each user. In principle, network latency can even be reduced, by pre-fetching or pre-sending links that a recommendation system expects the user to select.

Assume that there is an unknown matrix M of users $\times$ objects: each entry represents some user's true level of interest in some object. We generally know some entries of M (perhaps imperfectly). For example, an ISP knows how long its customers have spent reading various Web pages, an online advertiser knows who has clicked on what ads, and a supermarket can track each customer's purchases using frequent-shopper cards.

In patents co-authored with Fred Herz, Lyle Ungar and others, I have investigated methods for estimating the full matrix M. Each user's favorite objects can then be identified and ranked. A wide set of statistical techniques is available: clustering, k-nearest neighbor, principal components analysis, hierarchical Bayes, decision trees, neural networks, and so on. Such techniques can yield substantial real-world benefits. In a commercial deployment that substituted our personalized recommendations for generic ones, CDNow, a leading online retailer of compact discs, increased its sales from email recommendations by over 100%.

Many of these techniques rely on defining similarity metrics: similar users are then assumed to like similar objects. Collaborative filtering (Maes 1994) defines two users to be similar if they like some of the same objects, and two objects to be similar if some of the same users like them. We have improved this notion of similarity by also considering information not in the interest matrix M. Thus two users may also have common demographics, and two videos may have common actors. An especially useful case—well-known in the information retrieval community—is that two texts may have common vocabulary or style. **Textual similarity** is applicable not only to books and Web pages, but to any objects for which written descriptions or reviews are available online.

This technology raises a host of interesting implementation issues—notably privacy and distributed computation, where I have developed some protocols. We are also investigating other applications of the technology. We are particularly interested in direct applications to information retrieval and in facilitating browsing of large collections of documents or other objects. Another promising area is message routing—e.g., efficient multicast, filing of incoming messages, and automatic routing of e-mail queries to an appropriate expert or public forum. We have received patents on many of these ideas.

Future Work

Over the next five years, I plan to continue conducting basic research on problems where the input is unrestricted written language. The challenge is the pervasive ambiguity of human communication. My approach is to efficiently choose the most *probable* interpretation of the ambiguous input, under a carefully constructed, linguistically sensitive probability model.

I am especially keen to address the weak point of this approach—its heavy dependence on expensive training examples (input-output pairs). Two interlocking strategies can reduce this:

- Extract value from the abundant, multilingual online supply of *raw* text (newsfeeds, web).
- Extract *more* value from available input-output pairs, as I do in my thesis. This involves identifying new linguistic features that help predict the output; cleanly integrating them with other evidence in a formal statistical model; and developing parameter estimation and optimization algorithms for that model.

These two strategies can be combined to do reestimation or EM, where one repeatedly feeds raw text to an automatic system and then uses the output to retrain the system. If we retrain at each step using only the outputs where the system was most confident, the system will stretch gradually, moving to embrace those new examples that it can just barely handle.

New problem domains that I hope to study include machine translation (MT) and information retrieval (IR), as well as subproblems such as word sense disambiguation (WSD). Because statistical optimization ideas can be applied to a wide range of problems, both practical and theoretical, there are plenty of opportunities for collaboration with students and others.

Finally, in the world of applied research, I am interested in embedding linguistic and statistical techniques into user interfaces. A key goal is to automatically discover categories and relationships in large collections of documents, using all the available data: text content, text style, document layout, storage location, hyperlinks among documents, user queries, and access logs. The attendant goal is to use the inferred structure to help users browse, search, and conceptualize the document collection; to recommend documents; and to suggest new connections among related documents.