A Whirlwind Tour of Automated Language Processing for the Humanities and Social Sciences

Douglas W. Oard

Abstract

Automating some types of language processing holds great promise for helping us develop new ways of drawing insight from the world's linguistic legacy. But "promise" has many meanings, and this is a promise that has not yet been kept. This essay outlines the structure of the relevant disciplines, briefly describes the process by which automated language processing systems are created, and then offers some suggestions for how systems that better meet the needs of humanities and social science scholars might be built.

Introduction

We find ourselves at the threshold of a new era. Behind us is an era of almost entirely manual markup and transcription; ahead we envision increasing reliance on automation for at least the more mundane parts of that work. We regularly hear impressive claims for what future technology—always, it seems, *future* technology—will be able to do for us. Why is this future perpetually just over the horizon? The reason, I argue, is simple: those who could build these marvels don't really understand what marvels we need, and we, who understand what we need all too well, don't really understand what can be built. So we find ourselves in a situation a bit like the one depicted in the old cartoon of a blind person ringing the doorbell at the school for the deaf: we need new ways of communicating. Learning more about the other folks is a good way to start any process of communication, so in this brief essay I'll share a few of the things I have learned in my time among system builders. The situation is really quite simple: they are organized as tribes, they work their magic using models (rather like voodoo), they worship the word *maybe*, and they never do anything right the first time.

The Many Tribes of Language Processing

We seem to lack the right vocabulary for talking about this subject. Some refer to the broad subject as “text mining”—a term that has been used in so many incompatible ways that it may be better suited to marketing than to research. The core challenge here is social rather than technical: research communities form in ways that tend to balkanize the intellectual space. Rather than fight it, let’s go with the flow and look at these communities in the ways that they think of themselves.

As a first step, it would be helpful to say a word about the four forms of human language. Four? Yes, four. Spoken language, written language, and sign language probably immediately come to mind. But what’s the fourth? It is character-coded language, by which I mean what some call e-text: digital representation of individual characters (for example, English text represented as a sequence of ASCII characters). While this is indeed just another form of writing, the distinction is an important one because other forms of human language must generally be converted into character-coded text before we can easily manipulate their content. This distinction then serves to define two very active conversion communities: document image processing and speech processing. (Automatic transcription of sign language is not yet nearly as well developed.)

Like me, you probably grew up referring to document image processing as “OCR.” Optical character recognition (OCR) is indeed an important part of the process, but it is just one piece of a complex pipeline that starts with what might generally be termed “layout analysis.” The goal of layout analysis is to reconstruct the logical structure of a document. You might think of this as an attempt to recover the structural markup from which the document could have been generated. This is usually a three-stage process: (1) detect the physical structure (e.g., where on the page was that handwritten annotation made?); (2) classify each item using meaningful categories (e.g., logo, salutation, or body text); and (3) infer the logical structure from the available evidence (e.g., use relative position to guess which part of the text a handwritten annotation refers to). As you can see from this example, document image processing is about more than recognizing the correct sequence of printed characters: we need to handle handwriting, logos, structural elements such as tables and captions, and quite a challenging set of inferences about the author’s (or annotator’s) intent. As we will see again below, issues beyond mere content are also sometimes important. Can we tell from the style of the handwriting who wrote this note? Can we reliably determine what type of document this is (a form? a business letter? a memorandum? a page from a book?). All these problems are familiar to humanities scholars. If only they were equally familiar to our OCR programs how much easier our lives might be. Researchers who work on this gather each January at the Document Recognition and Retrieval Conference in San Jose, California. If you want to study document image processing engineers in their natural habitat, that’s the place to be.

Similarly, speech processing involves far more than the "automatic speech recognition" (ASR) that we all have heard about. There are essentially three subcommunities within speech processing: (1) interactive voice response systems (like the ones that answer the phone when you call an airline); (2) individually trained dictation systems, which were the first system to reach the market; and (3) systems that are still in the research lab. Research systems will be of greatest interest to us, since applications such as transcribing interviews, meetings, or streaming media require that we be able to accommodate a great deal of variability. Often the first step is to automatically figure out who spoke when, which goes by the unfortunate name "diarization." Once we know that, then transcriptions can be automatically adapted to do as well as possible on each speaker. This is followed by disfluency repair (e.g., to get the "umms" out) and then by (also infelicitously named) "pretty printing" techniques that guess where to insert sentence boundaries and capitalization and that try to convert spoken numbers to a reasonable written form. Speech processing researchers can be found each year at the International Conference on Acoustics, Speech and Signal Processing (ICASSP).

You might think that completes our discussion of conversion since now we have character-coded text, but you would be wrong. A third important type of conversion is paraphrase: automating the conversion of one expression of a set of ideas in character-coded text to another expression of those same ideas. (Thankfully, intentionally changing the ideas usually still requires human involvement!) Two forms of paraphrase are of particular importance: summarization and machine translation. In summarization, we seek to express some part of the ideas more succinctly. So-called *extractive* summarization techniques do this by simply selecting some parts of the text to show you—Google search results are one familiar example. You'll be disappointed to learn that that's pretty close to the state of the art—which provides some measure of job security for the people who write abstracts, I suppose. Summarization researchers can be found at the Text Analysis Conference (TAC), held each year in Gaithersburg, Maryland.

The other key type of paraphrase, *machine translation* (MT), works essentially like a translating parrot: the machine "hears" one language and tries to parrot back those ideas using words from another language. Because different languages might put their words in a different order, this is a really challenging problem that keeps MT researchers up late at night. As anyone who has used one of the many free Web translation services knows, the results are sometimes more useful for their humor than for the elegance of their expression: nuance is not the machine's long suit. You can study machine-translation researchers in the wild (along with their friends from natural language processing) at the annual conference of the Association for Computational Linguistics.

In some sense, all of this is natural language processing (NLP), but rather early on that moniker got appropriated by the people interested in telling nouns and verbs apart (remember diagramming sentences during your grammar school days?). Over the years, the NLP community (who also call their field "computational linguistics," which has a bit more of an academic ring to it) grew to embrace several large-scale problems, including summarization and MT. Three others are particularly noteworthy: extraction, classification, and clustering.

Extraction is the problem of identifying spans of text that are important for some purpose. The canonical example in NLP is to find proper names (e.g., names of people) in newspaper stories. But it doesn't take too great a leap of imagination to realize that we might use similar techniques to at least partially annotate much of what we call "coding" in the social sciences, i.e., labeling the things that our informants say with our interpretation of their meaning. This requires that we combine extraction with the second key capability: classification. The canonical classification problem is that I show you 100 newspaper stories and I tell you the category to which each belongs (international news, finance, sports, etc.). I then show you story number 101, and you decide which category it should be assigned to. When extraction and classification are combined (now classifying the span of text, not the entire story) the result is called "tagging" (which is unfortunately confusable with the more recently introduced idea of "social tagging," in which we trick ordinary people into doing a similar kind of work for us). Showing the machine all those examples is a bother, so clustering, the third key capability, tries to avoid that by just assuming that things that are similar should be labeled in the same way. Of course, that doesn't tell you what the label should be, but extraction might help with that (just extract whatever words seem to be most strongly associated with the cluster and hope for the best).

As this brief description has illustrated, these three capabilities can be put together in different ways for different purposes. Some well-known examples are authorship attribution (a type of nontopical classification), duplicate detection (a restricted form of clustering), and creation of a concordance (which is simply clustering text spans that share a common term). There are, therefore, many reasons why hanging out with NLP folks can be a good use of your time.

The black sheep of the NLP family is information retrieval (IR), which is a fancy name for what the rest of us call "search engines." IR and NLP developed as separate fields because they initially had little in common; IR folks just want to build useful systems without worrying too much about linguistics, while NLP folks start with linguistics and work toward useful systems. The two communities have much in common, and indeed you can find work on classification and clustering in both places. But search engines never did get subsumed into NLP, so you'll need to go to an IR conference if you want to hear the latest about searching. Interestingly, the IR community itself is somewhat bifurcated, with the IR systems folks hanging out with each other at the annual conference of the Association for Computing Machinery (ACM) Special Interest Group on Information

Retrieval (SIGIR) and the human-centered side of the field most in evidence at the annual conference of the American Society for Information Science and Technology (which is not really as U.S.-centric as it sounds, but it makes for a clever acronym).

What can we conclude from this techno-smorgasbord? One fairly obvious conclusion is that we need to find ways to communicate across disciplines about what is needed and what can be built. When such disparate worlds meet and try to communicate, they often select "boundary objects" that both can understand. In this case, we call that boundary object "metadata," and that is where we next turn our attention.

Mastering Their Voodoo

We are ambivalent about our metadata. People often misunderstand "ambivalent" as expressing a lack of preference; more properly, it means that something possesses both good points and bad points. Indeed, that's a reasonable summary of how many people feel about metadata. We like metadata because it allows us to get at meaning and the context in which that meaning arises, not merely at how that meaning was expressed in some specific case. Builders of language technology would say the same thing by observing that metadata allows us to go beyond the "surface form" to expose "latent variables": that way of saying it better fits their way of thinking about "models" that contain "variables." But metadata introduces its own problems; among the most frequently mentioned are cost and consistency. Interestingly, technologists are not nearly as bothered by these problems as we are, in part because they already understand that what we are trying to do is impossible.

OK, that's a pretty strong claim, so it probably merits a bit of discussion before we go on. Two factors combine to prevent us from creating perfectly accurate metadata. First, we don't always know for sure what the texts we are working with really mean. Second, we don't always know for sure what the metadata that we are creating really means. Solve those two problems, and this would be easy.

But the issue is not that we don't know how to solve these problems; it is that we know they *can't* be solved. Let's start with the question of what a text means. Language is a human creation, and language use is a creative act. Indeed, it is our ability to reason in the presence of ambiguity that makes it possible for us to express new ideas using an existing language. But wait, you say, isn't well-structured metadata supposed to allow for that? Here we meet the second problem: we simply can't agree on what we mean by our metadata. Consider a very well-standardized classification scheme, perhaps one that could be applied to this paper. Then run a quick thought experiment: train 1,000 indexers to classify essays like this one without showing them this paper. You know full well that no matter how well they are trained, and no matter how careful they are, some of the indexers will disagree with others about how this essay should be classified. The reason for this disagreement is not something that we can change, because the true meaning of our metadata exists only in our minds. Assignment of metadata is always an expression of an opinion rather than a statement of fact. Since people naturally will sometimes hold different opinions, our metadata is bound to exhibit some degree of inconsistency.

These same problems plagued NLP researchers for decades because NLP was originally conceived of as first encoding meaning in ways that people could understand, and then using that encoded meaning to do something useful. Starting in the 1980s (and with roots that go back further than that), a group of young turks in the NLP world decided to simply stop worrying about all this and learn to love uncertainty. When asked whether statement *A* has meaning *B*, they would always answer "maybe," and then work some wizardry with probability theory to figure out just how likely it was to be true. This proved to be a bit of a niche industry in the NLP business until some of the young turks demonstrated an MT system that did as well as the best existing systems by using statistics with only three facts: spaces separate words, periods end sentences, and an awful lot of examples of what good translations look like are available. It was this third fact that changed everything. When examples of language use were scarce, the human ability to see broad patterns from a few examples provided a useful foundation for NLP. But once computational access to language became ubiquitous, the ability of the machine to identify and memorize exceptions rapidly outpaced human abilities. And this is what made it possible for probability theory—the "science of maybe"—to come to the fore. Indeed, the transformation has been so complete that statistical modeling now lies at the core of every one of the disciplines identified in the previous section.

This tectonic shift has two important implications for us: we must learn new ways of thinking about what we are doing (generating and using metadata) and how we are doing it (using computational models). Jeannette Wing, who directs computer science research at the National Science Foundation, refers to this as "computational thinking," and she claims that it can be good for you regardless of whether you have any interest in computers. Let's take this one piece at a time, starting with computational modeling.

The word *model* is usually defined as a representation of some aspect of reality. Computational models often focus on behaviors, specifying how some input is related to some output (the classifiers mentioned earlier are one example of this). Over the years, the document image processing, speech processing, NLP, and IR communities converged on what is generally referred to as an "evaluation-guided research paradigm." The key idea here is that they start by identifying some challenge problem (e.g., a set of newspaper stories and a set of category labels to be assigned to those stories), an answer key (a "correct" set of assignments), and an evaluation measure (e.g., what fraction of the system's assignments are "right"). The programmer then goes off and designs a system that does the job, albeit not perfectly. After seeing the results, the programmers go back to the lab, try to build a better system, and again examine the results. They repeat this process until they run out of ideas. Because these systems are just trying different ways of learning the associated probabilities, the process can be partly automated, and it is not uncommon for developers to try a hundred, or even a thousand, variants of their system design overnight. This process has proven to be remarkably effective, but it has one key weakness: if the developers can't measure it, they can't improve it. So the entire process turns on how the challenge problem, the answer key, and the evaluation measure are constructed. The good news is that scholars in the humanities and social sciences don't need to learn probability theory to help guide this process. But we do need to start creating challenge problems, answer keys, and evaluation measures that reflect what we actually need the technology to do. So find someone who does this kind of research and ask that person to describe the challenge problems that they're presently working on. You'll be appalled by how far those "canned problems" are from what we really need. No wonder this stuff doesn't work well for us yet: the developers of the technology we need are not yet asking the right questions.

The bad news, however, is that humanities scholars are going to need to learn a bit of probability theory (many social scientists will have a leg up there). The reason for this comes back to the weakness in our boundary object—the way we think about metadata. When we've asked, "What metadata should be assigned here?" we have really meant, "What is the probability distribution over possible values for the metadata that should be assigned here?" We just didn't know that's what we meant. I am realistic enough to realize that we are not all going to go out and study probability theory just so that we can understand what all those computer scientists are saying. In the near term, this is why we need to work in interdisciplinary teams, learning from each other. But just as the children of "digital immigrants" grow up today to be "digital natives," our graduate students will grow up in a brave new world in which the answer to every question is "maybe" (assuming that they can keep a straight face with us long enough to pass their dissertation defense). So when I say that we need to learn probability theory, I don't really mean you and me—I mean our students. But nothing could be more natural; we merely need to shape the world in which they can do it.

Getting It Right

Peter Drucker once observed that the best way to predict the future is to create it. So let me close this essay with a few thoughts on what I think we should do.

- *Build useful tools, but don't try to automate the intellectual work of scholars*. This may seem obvious, but that hasn't stopped people before who have tried to build machines that do things we don't yet understand.
- *Dream big*. It is tempting to think about how best to use what we can already do (as the many studies that we already have that are

based simply on counting words amply illustrate). But real progress will come from the intersection between envisioning what we need and understanding what can be built. We're not going to get there if we keep starting with what has already been built. The key to the future is what we can model, not merely what we can see.

- *Waste money wisely*. After people landed on the moon, the phrase "it's not rocket science" entered our lexicon as a way of explaining that something wasn't really as hard as it might seem to be. But the challenge we face is not rocket science: it is harder than rocket science. After all, rocket scientists know what they are trying to do; they just need to figure out how to do it. We, by contrast, need some way of learning about what we are really trying to do. I used to work for the chief of naval research, who once said in a speech, "I am the only admiral in the Navy who can be wrong 90 percent of the time and keep my job." Why? Fundamentally, because technology researchers don't really know what it is they are trying to do. So initially (and, quite often, repeatedly), they do the wrong thing. The good ones learn as they go, and in the end they do some right thing, even if it was not really what they were trying to do in the first place. Essentially, this is the culture of the inventor, and it is one that we would do well to learn a bit more about. This may be our most challenging cultural shift, but it is one that we must make if we are going to make progress for one simple reason: metadata is not the right boundary object. The natural boundary object around which to build a conversation about what can be built is the system that creates that metadata.
- *Don't reinvent the wheel*. When you come down to it, statistical language processing is all about learning from examples. When people started thinking this way, it was natural to start by hand-building examples. For example, when people wanted to automate the process of drawing sentence diagrams (which they call "parsing"), they hired a slew of people to spend a few years generating some sentence diagrams that their machines could learn from. The leading edge these days, by contrast, is focused on taking advantage of examples that already exist. For example, when Ed Hovy wanted examples of good summaries for a week's worth of newspaper stories, he looked to a weekly newsmagazine. What does this have to do with us? Well, we have been building examples of what we need for some time. The trick is to think of the things that we have already marked up as "training data." Just tell someone who works on statistical language processing that you have heaps of training data already created for a new problem that is of great importance to our society. A sure ticket to instant popularity.
- *Make friends*. We're like yin and yang: we have the problem and they have the solution, so we need to find ways to work together. As a first step, there are now workshops at some of the conferences mentioned above that often go by names like "Cultural Heritage Applications of Language Processing." That's a springboard that could ultimately lead to formation of project teams, but only

> if we start going to their workshops (or they start coming to ours). Our European colleagues are ahead of us here: they've been putting money on the table to support interdisciplinary project teams that will work together for a few years on a specific problem. Of course, we do some of that in the United States as well—perhaps fewer teams, but sometimes with more resources per team. This is a natural approach, but we should think of it as a means to an end rather than as the end in itself. The byproduct of projects like this is a new cohort of doctoral students who will be the "natives" in this new world. The first generation of our young turks is already in place, and that will make the path that much easier for the next generation. These students are without question our future. Dan Goldin, a former administrator of National Aeronautics and Space Administration (NASA), had a mantra of "faster, better, cheaper." Ultimately, NASA decided that it could have any two of the three, but not all three, and today someone else runs NASA. But Goldin's idea was the right one: if you change the way you think, you can sometimes get all three. And the shift from interdisciplinary teams to interdisciplinary scholars will likely be such a transition. Nothing we could do is more important than educating the next generation of scholars to work at this intersection.

For many years, our technology colleagues have built provocative demonstrations of what they can accomplish. That is the "field of dreams" approach, and it is the only practical place to start: if they build it, (maybe) we will come. The ball is in our court.

Acknowledgments

This work was supported in part by National Science Foundation awards IIS-0122466 (MALACH) and IIS-0729459 (PopIT). I am grateful to my colleagues on both projects for the opportunity to learn from them, but they will be pleased to learn that the opinions expressed here are mine alone.