MACHINE LEARNING AND AUTOMATIC LINGUISTIC ANALYSIS: THE NEXT STEP

Eric Brill

Department of Computer Science
Johns Hopkins University
Baltimore, Md. 21218
brill@cs.jhu.edu

ABSTRACT

In order to continue building systems with progressively more complex natural language capabilities, it is crucial that great strides are made toward solving the core linguistic analysis problems for complex and possibly unrestricted domains. A great deal of progress has been made by applying machine learning techniques to automatically train systems from manually annotated corpora to provide detailed linguistic analyses to sentences. This paper examines a number of issues within this paradigm of automatic linguistic knowledge acquisition and how they relate to pushing progress in the field of natural language processing over the next decade.

1. INTRODUCTION

As the fields of language and speech processing continue to progress to more and more complex problems, the need for core linguistic analysis technology, such as lexical disambiguation and phrase structure analysis, is going to become crucial. When building a system for a single constrained domain, such as weather reports or ATIS, it is not clear that employing a generic natural language analyzer is an effective approach. Given sufficient time, one can build a domain-specific semantic-flavored annotator, taking into account the particular idiosyncrasies and likely strings in that domain. But in trying to build a system where the input is not restricted to be from one particular domain, the need for more general language-processing algorithms becomes clear.

To determine what research the field should be conducting to best facilitate the deployment of sophisticated language processing systems over the next decade, we must ask two questions:

1. What are the current bottlenecks: where is progress most needed in core linguistic analysis technology?
2. How can we push the state of the art in these essential technologies?

The answer to the first question might seem obvious: part of speech tagging, word sense disambiguation, parsing, pronoun resolution, etc. However, it is not clear what level of granularity is most useful nor what degree of accuracy must be achieved for the different core linguistic analyses to be useful. For instance, parsing could range from just identifying noun phrases to a very detailed parse including information about traces and other abstract linguistic entities. For part of speech tagging, one could use a coarse tagset containing only the tags NOUN, VERB and OTHER, or a much more refined tag set containing hundreds of unique tags and making rather subtle distinctions. While solving the harder problem will also solve the easier problem, it will be much more difficult to develop programs that solve these problems at a very fine level of granularity. Therefore, we must carefully determine at what point the value added in finer granularity and higher accuracy does not justify the increase in difficulty in trying to solve the problem.

The last few years have seen a large boom in machine learning approaches to natural language processing, with algorithms being developed to automatically extract linguistic information from corpora, either in conjunction with or instead of manually deriving such knowledge. Along with employing machine learning algorithms, the field has also adopted the machine learning paradigm of measuring the efficacy of an algorithm by splitting properly labeled data into disjoint subsets, one of which is used for training and the other for testing. There are now readily available data sets for many core linguistic analysis problems, including part of speech tagging, word sense disambiguation and parsing [MSM93, DeM90, Mil90], and annotated corpora are becoming available for a wide array of domains and languages.

This shift to training and testing on linguistically preannotated corpora has impacted the field in a number of positive ways. For one thing, the availability of these resources has helped shift research away from small toy problems

This work was supported in part by NSF grants IRI-9502312 and IRI-9618874

to larger-scale problems involving naturally occurring sentences. It is no longer possible to publish a paper describing a system that can analyze the sentence **John loves Mary.** Instead, people have begun working in earnest at developing systems to analyze naturally occurring sentences, sentences that really were spoken in a conversation or that really did occur in a newspaper. The existence of these corpora also makes it relatively easy to compare the performance of different algorithms and allow the field to assess what aspects of an approach are responsible for performance, something very difficult to do when there are just as many definitions of a problem as there are attempts at solving it.

In both natural language processing and speech recognition, we have seen the many benefits of having a community work on common tasks. Therefore, it seems that the right approach is to continue developing corpora manually annotated with linguistic structure, thereby providing common training and test sets for researchers to use in attempting to solve these problems. In addition to measuring accuracy, these corpora would be used as training corpora for machine-learning algorithms. However, before continuing along this line, it is important that we better understand the paradigm of using annotated corpora for training and testing in natural language processing. Below we will first examine the basic tenets of corpus-based natural language processing. Next we will address the question of determining where progress is most needed.

2. WHAT IS AN ANNOTATED CORPUS?

In order to test the applicability of a machine-learning algorithm to a particular problem, one takes a set of problem instances and divides them into a training and test set. A problem instance consists of a vector of feature values and an instance label. For example, if we want to build a system to predict from a faculty candidate's application how much grant money he or she is likely to bring in, we would first decide upon a set of features that may correlate with grant procurement. Then a sample could be taken of faculty members, assigning the proper set of feature values extracted from their applications and setting the instance label to be the amount of grant money that faculty member has received. Different machine learning algorithms could be compared by having them predict the grant yield for a set of faculty members from their application data, and determining which algorithm is more accurate.

We can take the same approach to building a program to learn how to linguistically annotate word sequences: collect a set of sentences, manually label each sentence with the linguistic information we wish to learn, and then on a test set see how close the output of the trained system is to the truth. This is exactly the approach taken in corpus-based natural language processing.[1] However, there are a number of crucial differences between the linguistic knowledge acquisition example and the faculty applicant example. These stem from the fact that the faculty applicant example consists of real-world measurements of actual things and provides output that is clearly useful in and of itself.

To linguistically annotate a corpus, one derives a representation and a set of guidelines for appropriately assigning a structure consistent with that representation to any string of words. These guidelines are usually based upon a number of factors, including: linguistic principles, annotation usefulness and the ability of a human annotator to consistently follow the guidelines. The end result, one hopes, is a linguistically sound annotation, representing a meaningful description of certain linguistic properties of the string. However, a corpus is a human-made entity, and as such the annotations are a combination of the underlying target representation and human foibles. Although manually annotated corpora are now ubiquitous in NLP research, there has been very little research addressing just what an annotated corpus actually is, how it is best used, and what its inherent flaws are.

2.1. Corpus Consistency

Since unlike our faculty funding example, the labels in a linguistically annotated corpus are much more subjective, we need to be able to gauge how consistently the corpus is annotated. We can define the consistency of a corpus as how often two entities (e.g. words or phrases) are given the same annotation when these two entities appear in linguistically equivalent environments. For instance, one would hope that the subject noun phrases of the sentences:

1. **The three large birds in the tree** had been sitting there all day.
2. **The seven little crows in the tree** ate all of our cabbage.

should be assigned identical syntactic structure, as they are syntactically invariant. Note that in the funding example, it is perfectly fine to have inconsistent instances, for example two faculty members who had identical CVs as applicants but who now bring in dramatically different amounts of funding. The reason is that with the faculty members we are taking real-world measurements, and so inconsistencies, if they occur, are just facts of the world. A linguistic annotation is, one hopes, a human approximation to an underlying linguistic structure that is not visible and not yet well-understood. If we see different labelings of two identical instances, this is problematic.

[1] For a recent overview of this field, see[Be97].

As a simple exercise, we ran the following experiment. We took the tagged and parsed Penn Treebank Wall Street Journal Corpus. In that corpus, there are 196 sentence types that occur more than once. Duplicated sentences account for approximately 2% of the total sentence tokens in this corpus. We then compared the manually annotated structure of these duplicated sentences. In cases where a sentence appeared more than twice, we randomly chose two instances of it. Of these pairs, 32% matched exactly in terms of their annotations. For sentences of length less than 10, 42% matched exactly (47/112). For sentences of length 10 or greater, only 19% matched exactly (16/84). Since consistency is not achieved even for the most stringent cases of linguistic environment invariance, it is doubtful that consistency is achieved in more subtle cases of environment invariance.

One might argue that having less than 100% consistency just means that the upper bound on the accuracy one could hope to achieve will be less than 100%. However, this is a potential trouble sign that needs further study. It could indicate that a corpus is more a recorder of human annotator behavior than a pure reflection of hidden linguistic structure. For instance, it could hypothetically be the case that that noun phrases in short sentences are manually annotated with more internal structure than those same noun phrases in longer sentences. Such a phenomenon could be attributed to the fact that shorter sentences present less of a cognitive load on the human annotator, and therefore they are annotated more carefully and in greater detail. This would result in linguistically inconsistent annotations. But this does not just (and in fact may not at all) lower the ceiling on what accuracy a parser can hope to achieve. A learning algorithm that either explicitly or implicitly had the length of the sentence as a feature or constraint could circumvent this annotation inconsistency problem by basically learning the behavior of the annotators, thereby achieving higher accuracy than a system that does not learn this. This is problematic, because surely it is linguistic structure we wish our system to learn and not human annotator behavior.

There are likely to also be fairly significant differences in the annotation behaviors of different annotators. In [Rat96] it was shown that for the part of speech tagging of the Penn Treebank there are significant stylistic differences in how different annotators dealt with various linguistic constructs. Indeed, if one had access to the name of the annotator who annotated a particular sentence as a feature, one could likely significantly enhance performance, again an indication that the annotated corpus is really a combination of linguistic truth and human annotator behavior.

2.2. Implicit Bias

Many manually annotated corpora are created using a bootstrapping method, where first a small amount of text is manually annotated, then this text is used to train an automatic annotator. People then annotate additional text by correcting the output of the automatic annotator, and then the program is retrained using this additional material. This means that the corpus will have a bias induced by the particular automatic annotation program that was used. Correcting the output of an automatic annotator is likely to give a very different flavor of annotation from manually annotating from scratch. This is because the default action in the former case is to accept the annotation provided by the program, whereas in the latter case every annotation decision is made from scratch. Although it is likely to be a much more efficient means of generating an annotated corpus, it will reflect the properties of the underlying automatic annotator. In fact, such a method could minimize corpus inconsistencies, since all annotators will be working off of the same basis annotation. But again if our goal is to learn the annotations in the corpus, we have to question to what extent we are learning linguistic versus nonlinguistic information.

To give a concrete example, many part of speech tagged corpora, including the Penn Treebank, were created using a Markov-model tagger to tag the text and then having people correct the tagger output. People have since run experiments training and testing Markov-model taggers on these corpora and have achieved impressive rates of tagging accuracy. But is this because they are learning useful linguistic information, or because they are reflecting the underlying Markov-model bias of the corpus, stemming from the way it was constructed. Indeed, the underlying Markov-model bias also to some extent induces in the corpus the property that local information is sufficient for deciding the tag of a word. A vast number of learning algorithms have been successfully applied to part of speech tagging, all relying on local cues for disambiguation. We have to wonder to what extent the success is attributable to actual learning of useful linguistic patterns versus being a reflection of the way the corpus was constructed.

There is yet another potential form of implicit bias in a manually annotated corpus, stemming from the annotation guidelines. While the guidelines lay out linguistic rules that the human annotators are to follow, there are likely to be many non-linguistic specifications as well. For instance, in the annotation style manual for one particular parsed corpus, it specifies how prepositional phrase attachment is to be annotated in cases when the person cannot determine the proper attachment. The manual specifies a default action: if the annotator cannot determine the appropriate attachment then they should attach it to the closest (lowest) constituent. This means that if we build two parsers that are identical except that one attaches low as a default when all constraints or rules fail to apply, that parser will perform better. However, it is not doing a better job at capturing linguistic information, rather it is doing a better job of mimicing a partic-

ular nonlinguistic property of the corpus which arose from an arbitrary specification in the annotation guidelines.

3. WHAT IS ANNOTATED CORPUS MIMICRY GOOD FOR?

The immediate goal of corpus-based machine learning of natural language is to learn how to mimic the structure of a manually annotated corpus as closely as possible. We have discussed above some of the problems of this research paradigm, namely that a linguistically annotated corpus is only partially a linguistic entity. Even after we settle the many issues related to learning from annotated corpora, we are still faced with a serious question: if one could create a program that could perfectly mimic the human corpus annotators, what would it be good for? While most people studying computational linguistics believe that the ability to accurately annotate text with such information as parts of speech, word senses and phrase structure is essential for sophisticated natural language processing, many questions remain. Just what are such annotations good for? What can be done with a very coarse-grained annotation; what additional tasks could be done if the annotation were fine-grained, capturing subtle linguistic distinctions? What level of accuracy is needed for these annotations to be useful for various tasks?

To attempt to answer these questions, the research community could choose a number of tasks, such as information retrieval from natural language queries and answering SAT-style reading comprehension questions, and then provide a set of instances that have been manually annotated with very fine-grained linguistic information. This will then provide a test-bed to better understand what can be done once we are able to build programs that can output very accurate linguistic analyses, by exploring just how well one can do at solving these problems with such annotations being available. By decoupling research on automatic linguistic analysis from work on specific applications we hope to develop tools that are general enough to be useful for a wide array of applications. We can assure that this is the case by providing a range of corpora annotated with the linguistic analyses the community is working to develop automatic annotators for, and let the end users determine just how useful such linguistic annotations would be for their tasks.

4. CONCLUSIONS

To build systems capable of handling complex natural language input, we are going to have to make a great deal of progress in developing accurate core automatic linguistic analysis programs, such as lexical disambiguators and parsers. Automatically extracting linguistic information from manually annotated corpora seems to be a viable approach to solving these problems. Although such work has been going on for over a decade, we still know very little about just what a linguistically annotated corpus truly is, nor how to use these corpora to compare different automatic annotation programs. To make progress over the next decade, we first need to carefully assess and challenge the tenets of the field before proceeding further.

5. REFERENCES

[Be97] Eric Brill and Raymond Mooney (editors). Ai magazine special issue on empirical natural language processing, Winter, 1997.

[DeM90] C. DeMarcken. Parsing the lob corpus. In *Proceedings of the 1990 Conference of the Association for Computational Linguistics*, 1990.

[Mil90] G. Miller. Wordnet: an on-line lexical database. *International Journal of Lexicography*, 3(4), 1990.

[MSM93] M. Marcus, B. Santorini, and M. Marcinkiewicz. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2), 1993.

[Rat96] Adwait Ratnaparkhi. A maximum entropy part-of-speech tagger. In *Proceedings of the First Empirical Methods in Natural Language Processing Conference*, Philadelphia, Pa., 1996.