Is "Semantics" Just Picking the Right Syntax for the Context from Multiple Possibilities?

Breck Baldwin
Alias-i, makers of LingPipe
breck@alias-i.com

I have never come across a semantic analysis in software that didn't ultimately come down to a very syntaxy rule or bit of training data over syntaxy annotations. It is just the nature of what computers do--its all symbol manipulation in the end. But I do think there is an interesting distinction to be made that supports the workshop agenda: Syntax defines the space of possible analysis and semantics is the process of selecting the appropriate one from many possibilities. A consequence of this observation is that systems that optimize F-measure over first-best analysis are unsuitable for this approach. Instead n-best confidence based annotations are better.

For example, back when grammar writing was a pretty common thing to do for sentence parsing, the goal was to come up with all possible analyses for a sentence no matter how unlikely a parse. It was assumed that context, lexical preference or some bit of magic would pick the right parse. That still goes on today hidden behind machine learning approaches.

Applying the idea of "get all possible" analyses speaks to the idea of recall or coverage, with the semantic processing producing the precision or accuracy. I have taken the position that the best way to get semantics into the picture is to start with 100% recall for the semantic annotation and then see how good the precision can get.

Consider something as simple as named entity identification for a single category--genes. A base line implementation that produced all substrings of length less than 12 words long labeled as gene gets 99.96% recall at 0.7% precision. Carpenter(1) showed that it was possible to get 100% recall (actually 99.99%) recall on gene names with 7% precision using confidence ranked output. Another order of magnitude improvement and we would be flirting with state of the art performance as measured with F-measure with a very interesting recall.
A consequence of this approach is that system output will be contradictory over the same text stream. Downstream applications are forced to reason at the level of sets rather than a single analysis and even more interestingly we can have sets of analyses with associated probabilities.

An even more controversial position that I am willing to argue is that systems with less than 100% recall are fundamentally uninteresting from a "solve real problems" perspective and that the dicotomy between syntax and semantics becomes less pronounced. The question ought to be "what is your precision at 100% recall" rather than F-measure.

http://www.colloquial.com/carp/Publications/biocreative-8-alias-i.pdf