Monotonicity Reasoning as a Bridge Between Linguistic Theory and NLI

This talk is (a) a report on ongoing work by a research community which takes inspiration both from linguistic theory and from computational linguistics; (b) a statement about the gaps between the two; and (c) a plea for further interaction. It is largely a position paper.

The topic of interest is *natural language inference (NLI)*, both on the computer and by humans. Although natural language semantics sometimes advertises itself as being a study of *inference* ("what follows from what?"), the study of inference proper is a topic that is hardly ever broached by semanticists. That is, just giving the truth conditions for sentences is enough of a task, and the semanticist in practice ignores inference. But the semanticist is not alone in doing so. People in several other fields could reasonably be expected to think about NLI, but they, too, do not take the matter seriously. Logicians are unable to cope with sentences "in the wild." Cognitive scientists are interested in human performance, but they usually confine their studies to syllogism-like snippets of two or three sentences. People in the NLI area of NLP build systems that can indeed learn by pattern-matching, but they rarely work on actual inference by humans.

Around 2018, the time that the BERT models burst on the scene, there were several groups of people working on *logic-based approaches to NLI*. These efforts did not use machine learning methods (at the time this did not seem like a great idea), but rather used some version of *theorem proving*, based on converting the assumption sentences and the conclusion sentences to some sort of representation and then calling a theorem prover. As it happened, the different logic-based approaches all performed about equally well on the datasets that were most important at the time, including the SICK dataset. But the BERT models out-performed all of them. So this is the backdrop of this talk.

Arrow tagging The main topic of the paper is monotonicity reasoning. This is an old topic in semantics, best-known perhaps from the work of Ladusaw, Dowty, Keenan, Barwise & Cooper, and others. When one pushes it further, a number of interesting issues and problems arise. In terms of the Bridges and Gaps theme, we have parsers for grammatical frameworks that have something approximating a clear mathematical semantics, and so

one could marry some a "algebraic" work on monotonicity with a parer that runs on text as it comes, and thereby get a very "lightweight" approach to inference. To be sure, the kind of grammar used in van Benthem's work was Ajdukiewicz/Bar-Hillel CG, probably the simplest kind of grammar, and one has to work a bit to extend it to a framework like CCG. That is, one has to "do the math", and that is one of the topics in this line of work. Skipping all of the details, we can now do"arrow tagging" on the computer. There are programs which input parsed trees from a CCG parser (see Figure 1), and change them a little, producing a parse tree with upward-looking arrows and downward-looking arrows, and other information besides (Figure 2).

Using arrow tagging in NLI To actually use arrow tagging in NLI in theory, one also needs a source of basic facts like $dog \leq animal$, and preferably an inference engine that could take temporary assumptions like *Sarah is a doctor* and add NP-level basic facts like $every\ doctor \leq Sarah \leq some\ doctor$. Arrow-tagging gives us a very "lightweight" theory of inference: build a good database of background facts, do the tagging, and make replacements. This will not cover for hard-core logical entailment, but it ought to be sufficent for everyday "unconscious" inference.

Connections to the "logicality of language area This talk will bring up a new point: the field of semantics developed by Chierchia, del Pinal, and others dealing with the "logicality of language" is also about monotonicity. So there should be some connection to the work by computational semanticists.

But can a linguistically-informed inference engine really compete with machine learning NLI? Yes, No, and Maybe. What we find is that monotonicity is a good strong first tool, perhaps unexpectedly so. One needs further theory to handle *syntactic variation*, and here, too, there are real connections to topics in semantics that are often neglected. One also can build *hybrid systems* that use both machine learning and arrow tagging. At the present time, the best performing system on the SICK dataset is such a system; it is called NeuralLog.

The overall theme of the talk is that the combination of machine learning + monotonicity reasoning is a going concern bridging the gap and leading to new problems and issues on both sides.

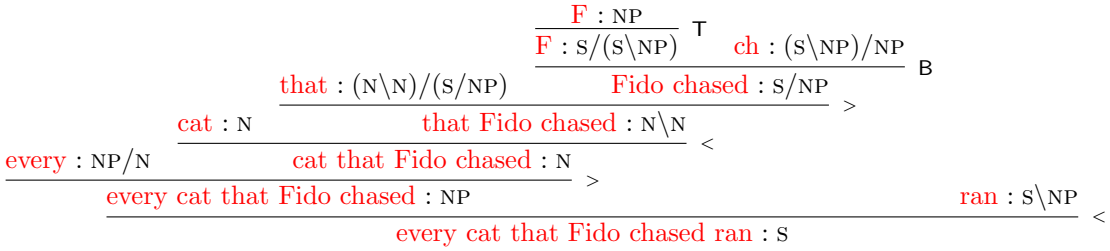

Figure 1: A CCG parse

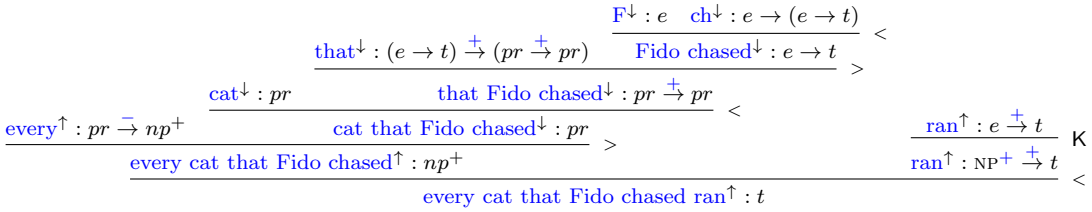

Figure 2: Arrow tagging in our CCG parse

system	P	R	acc.
majority baseline	–	–	56.36
ML/DL-based systems			
BERT (base, uncased)	86.81	85.37	86.74
Yin and Schütze (2017)	–	–	87.1
Beltagy et al. (2016)	–	–	85.1
Logic-based systems			
Bjerva et al. (2014)	93.6	60.6	81.6
Abzianidze (2017)	97.95	58.11	81.35
Martínez-Gómez et al. (2016)	97.04	63.64	83.13
Yanaka et al. (2018)	84.2	77.3	84.3
MonaLog + transformations	89.91	74.23	81.66†
Hybrid systems			
Hybrid: MonaLog + BERT	85.65	87.33	85.95†
Kalouli et al. (2020)	–	–	86.5
NeuralLog (full system)	88.0	**87.6**	**90.3**
– syntactic variation	68.9	79.3	71.4
– monotonicity	74.5	75.1	74.7