Incremental Constraint-based Equitable and Efficient Natural Language Parsing

Gregory Aist (Gregory.Aist@asu.edu)
Department of Computer Science and Engineering, School of Computing and Informatics,
Ira A. Fulton School of Engineering, Arizona State University, P.O. Box 878809, Tempe, AZ 85287-8809 USA

Keywords: Natural language understanding; pragmatics.

We present a novel computational method for realtime language processing, inspired by human language understanding, that is (a) incremental, (b) constraint-driven, (c) equitable to its information sources, and (d) efficient.

Previous incremental algorithms generally first construct a syntactic parse, integrating semantics and/or pragmatics later (Rose et al., 2002; Mori et al., 2001; Wiren, 1992). Even in cases such as categorial grammar (Steedman, 1991), with interleaved syntax and semantics and ready incrementality (Milward, 1995), pragmatic constraints are left to later processing stages. Even where pragmatic constraints are applied early, pragmatics is secondary information, used to modify parsing probabilities (Stoness et al., 2005) or to further constrain underlying syntactic rules (DeVault and Stone 2003; Schuler 2003). By contrast, we build neither syntax trees first nor semantic interpretations first nor pragmatic considerations first. Rather, all constraints are architecturally equal, so preferences emerge naturally.

In the following example the interpretation is driven by different types of constraints based on the information of each type that is available to understand the utterance. Consider two possible dialogues. Suppose there are landmarks (a rock); objects to place (a beacon, two flags); and the objects' color can be changed (red, yellow).

(1). A yellow flag goes near the rock.
(2). (a) Make the beacon red. (b) Make the flags the other color. (c) A yellow flag goes near the rock.

In sequence (1), no yellow flag exists when speech begins. Thus the word "yellow" invokes "yellow" as a noun and as an adjective; when the word "flag" arrives, further lexical and syntactic processing favors "yellow" as an adjective.

In (2), the color *yellow* – while not explicitly mentioned in the discourse – has been applied to two flags prior to (c). Prior to the onset of speech, the environment thus produces the pragmatic constraint *yellow* as a property, which in turn invokes the constraint that properties are described by adjectives. Together these constraints guide the *immediate* interpretation of "yellow" as an adjective.

The constraint representation includes *type* – word (w), category (c), etc. – and (*left*,*right*) boundaries. Constraint boundaries show how much of the utterance the constraint applies to. Boundaries can either be free variables or instantiated to a location in the input string. For example, a pragmatically induced constraint might have a variable for its left boundary and a variable for its right boundary, since it has not yet been associated with any particular substring of input words. Such constraints can be re-used as various possible input matches arrive, generating new distinct arcs in response to distinct input words. The original arc remains unchanged and the new arc may have instantiated endpoints if tied to a particular portion of the input. Constraints are incrementally added, as available, to a dynamic programming structure.

Our method is like chart parsing in not backtracking due to storage of partial results – except that arcs can have variable boundaries, and additional sources of knowledge act as constraints. It is like left-corner parsing in using both top-down and bottom-up constraints – except that it builds from more than just the left corner. It is like a blackboard architecture in using multiple knowledge sources – except that the constraints operate within an efficient unified framework and control structure (via dynamic programming) rather than being manipulated by arbitrary programs. All of the constraints could in principle include probabilities in order to allow estimation of the likelihood of the various analyses.

References

DeVault, D., & Stone, M. (2003). Domain inference in incremental interpretation. In *Proceedings of ICOS.*

Milward, D. (1995). Incremental interpretation of categorial grammar. In *Proceedings of the Annual Meeting of the European Association for Computational Linguistics (EACL 1995)* (pp. 119–126).

Mori, D., Matsubara, S., & Inagaki, Y. (2001). Incremental parsing for interactive natural language interface. In *Proceedings of the 2001 IEEE International Conference of Systems, Man, and Cybernetics.*

Rose, C., Roque, A., Bhembe, D., & Lehn, K. V. (2002). An efficient incremental architecture for robust interpretation. In *Proceedings of HLT.*

Schuler, W. (2003). Using model-theoretic semantic interpretation to guide statistical parsing and word recognition in a spoken language interface. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)* (pp. 529–536).

Steedman, M. (1991). Type-raising & directionality in combinatory grammar. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL 1991).*

Stoness, S., Allen, J., Aist, G., & Swift, M. (2005). Using real-world reference to improve spoken language understanding. In *Proceedings of the AAAI Workshop on Spoken Language Understanding* (pp. 38–45).

Wiren, M. (1992). *Studies in incremental natural language analysis.* Unpublished doctoral dissertation, Linkoping University, Linkoping, Sweden. (ISBN 91-7870-027-8)