

# Computing Textual Inferences

Lauri Karttunen

Palo Alto Research Center

Stanford University

A long-standing goal of computational linguistics is to build a system for answering natural language queries. An ideal QA system is able to determine whether the answer to a particular question can be inferred from another piece of text. For example, the system should recognize that the answer to a question such as Did Hillary win the nomination? is given by a sentence such as Hillary failed to win the nomination. None of the current search engines is capable of delivering a simple NO answer in such cases. But change is coming. Much progress has been made in computing textual inferences in recent years, much of it inspired by work presented at the Pascal RTE (Reconizing Textual Entailment) workshops.

Local textual inference is in many respects a good test bed for computational semantics. It is task oriented. It abstracts away from particular meaning representations and inference procedures. It allows for systems that make purely linguistic inferences but it also allows for systems that bring in world knowledge and statistical reasoning. Because shallow statistical approaches have plateaued out, there is a clear need for deeper processing. Success in this domain might even pay off in real money in addition to academic laurels because it will enable search engines to evolve beyond keyword queries.

The system I will describe in this talk is the Bridge system (a bridge from language to logic) developed at the Palo Alto Research Center by the Natural Language Theory and Technology group. I will first give a brief overview of the system and then focus on the way textual inferences are computed.

The inference algorithm operates on two AKRS (Abstract Knowledge Representation), one for a passage, the other for the question. It aligns the terms in the two representations, computes specificity relations between the aligned terms, and removes query facts that are entailed by the passage facts. If all the query facts are eliminated, the system will respond YES. If a conflict is detected, the system will respond NO. If some query facts remain at the end, the response is UNKNOWN. In some rare cases (John didn't wait to speak. Did John speak?) the response will be AMBIGUOUS indicating that the on one reading the answer is YES and on another reading NO.

The linguistic phenomena illustrated include hyponymy (hop => move), converse relations (win vs. lose, buy vs. sell), lexical entailments (kill => die), relations between simple predicates and their embedded complements (forget that S => S, forget to S => not S), and similar relations involving phrasal constructions (take the trouble to S => S, waste an opportunity to S => not S).

For a practical QA system logical entailment and presupposition are important notions but they are not sufficient to characterize all the inferences that a human reader will make. The Pascal RTE data contains many examples where there is no logical entailment between sentences but the annotators have indicated otherwise. In our ordinary conversation we make such "errors" all the time. John is very happy that he had a chance to read your paper does not actually entail John read your paper but in the absence of any contrary evidence, the hearer would certainly conclude that John had read the paper. Characterizing such "invited inferences" is an interesting challenge for semantic and pragmatic theory and essential for practical applications.