How Computational Linguists Handle Questions

Answering questions is a complex task for computational linguists. From a computational perspective, "[t]he question-and-answer process is interpreted as an information-matching game" (Harrah, 1961). The concept of relevance when analyzing answers is the new hot topic when it comes to erotetic reasoning. Marco De Boni argues for this idea of relevance and how to functionally apply it, while Kevin Knuth derives a logic that would determine if a question is a relevant response to a question being asked. These form some of the necessary functions required in making artificial intelligence, giving ai the ability to expand their knowledge base or supply answers from their knowledge base.

One of the major applications of these models would be for the use in text retrieval and context derivation of questions. In the past, closed domain question answering methods have been derived, but lack in their use cases. Boni goes so far to call them toy programs, due to their simplicity and limited scale of use. Rather, Boni and Knuth provide some techniques that would allow the development of a more comprehensive QA that could be applied in an open domain.

Before understanding how Boni and Knuth's theories work, it is important to understand how more basic concepts are utilized to perform the more complex tasks they propose. Tagging is an important tool for functionally breaking down questions. POS taggers allow for questions to be broken down into their relevant constraints, which are used when checking for relevance. Another important tool are corpuses, knowledge bases, and other large bodies of text. For open-ended systems, these are vital, as they are what holds the possible questions and answers they are being used on. Other important things required for erotetic reasoning include the basic models of logic used, like modal logic, predicate logic, first-order logic, and even very simple ones like boolean algebra, which Knuth makes heavy use of.

From here, we can understand more thoroughly what kinds of technology and schools of thought Boni and Knuth rely on. Boni prefaces his work by talking about past work in the field, and how their results were only applicable to closed-domain simplistic question systems. He then proceeds to explain why relevance logic would not suffice what he is trying to show, but rather there needs to be a new way of representing the complex structure of language. This is where he introduces his definition of relevance, and how it could play into determining correct answers. He claims that

> "the relevance of an answer is determined by how many constraints must be removed from the question for the answer to be proven; the less constraints must be removed, the more relevant the answer". (Boni, 2007)

He provides a few examples of how this applies to answers, by removing differing amounts of constraints in an attempt to show that the answers are relevant, but are not the exact answer, which goes to support his point that relevance is a sliding scale value, not a boolean yes/no. He also shows how an implementation would work in pseudo code. To tie his theory together, he uses a few Text Retrieval Conference corpii to use as datasets for his algorithm. He applies some rules that would remove specified constraints, and show that answers derived were of relevance to the question, showing that his methodology was sound.

While sentences are a valid way to answer a question, there are times when a question is unanswerable with current knowledge. In terms of a computer, this could mean a sensor isn't sensing something, or it has been asked to search for something it doesn't have information on.

A valid way to answer such questions would be in turn to ask a follow up question. For example, if someone were to ask “Why did the dinosaurs die?” and the knowledge base contained no definition of what a dinosaur is, the computer could ostensibly ask in return “What is a dinosaur?”. But how does it know if a question is relevant? Kevin Knuth proposes a solution to this. By modelling questions as a lattice of boolean statements, predicate logic can be applied, thus giving the ability to construct sub-questions from an overarching question as a means to solve issues with lack of knowledge. The math involved is based on “the algebra of questions, and its associated calculus, the inquiry calculus.” Knuth first defines a question as “the set of all logical statements that answer it”, which follows his argument. From here, he formally defines from the ground up the components of questions, questions themselves, and complex applications of the algebra that is able to be produced from the components. After defining his calculus, he discusses how this translates functionally: “This methodology promises to enable us to design machines that can identify maximally relevant questions in order to actively obtain information.” (Knuth, 2005) For self learning programs, like neural nets, this is the crucial step for the program to be able to both interact and grow its knowledge base, for without it, it is completely dependent on its corpuses of knowledge to be able to give relevant answers.

If both of Boni and Knuth’s theories were applied to a neural net, we could ostensibly see a natural language processing AI that was capable of open-domain question answering. This is a steep step, as it requires both the consumption of huge corpi to round out its initial knowledge base, plus some way to update the knowledge base based on the response the neural net receives from its exploratory questioning. However, this could unlock how to computationally analyze the natural language of questions, as has been attempted in the past, with programs like the famous SHRDLU by Terry Winograd. With these two theories, questions might be better understood from a computational linguistic standpoint.