COG 376 Michael Phillips
Final Paper 5/16/2014

Semantics & Pragmatics in Computational Linguistics

The Ambiguity of Meaning and the Need for Pragmatics

Computational linguistics is a field of linguistics focused on the modeling and formalization of natural language, so that language can be represented in a computational system. There are subfields of computational linguistics which aim to divide linguistics by its usual areas of study, such as phonology, morphology, syntax, semantics and pragmatics, and format the methods to specifically formalize each of these areas based on their own criteria. However, as we'll see, approaches used in the different subfields of computational linguistics tend to be similar in many aspects, and usually draw from the methodologies of overlapping ideas. In what follows, I focus on computational semantics and computational pragmatics. One of the biggest difficulties of formalizing language for a computational environment is dealing with meaning, because meaning in language, although we don't tend to notice, can be rather ambiguous, especially without context. First, I discuss semantics, and the approaches to semantic representation in computational linguistics, including the methods used, as well as the limitations. Second, I discuss pragmatics, and the approaches to computational pragmatics, including the methods used, as well as their limitations. The focus here is to discuss the representation of meaning in a formal language system, and what aspects of meaning make this representation so difficult (and pragmatics so relevant).

I. Semantics

First, let's define semantics as it pertains to linguistics (not necessary on a computational level). Semantics is the branch of linguistics that is concerned with the meaning of natural language. In particular, semantics deals with some level of signification in the language (a word, phrase, symbol(s), etc…) and their denotation.

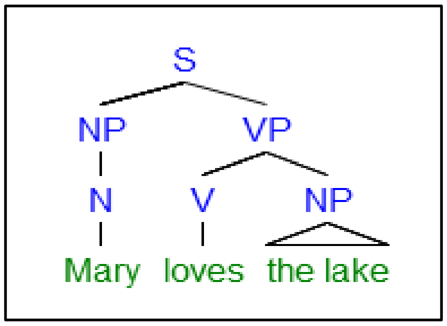
In computational linguistics, the goal is to formalize natural language so that a computer could deal with the information.

$$\begin{tikzcd}\text{Natural language} \arrow[r] & \text{Formal Language} \arrow[r] & \text{Computer Environment}\end{tikzcd}$$

Since semantics has to do with meaning, we must then find a way to formalize said meaning, in models, frameworks, or other forms of representation that could later be understood by a computer. So what are the approaches to semantic representation? The article "Computational Linguistics" published by the Stanford Encyclopedia of Philosophy recognizes three forms of semantic representation: (1) logicist approaches to meaning representation, (2) psychologically motivated approaches to meaning, and (3) statistical semantics.

(1) Logicist approaches to meaning representation

In this case, a "logicist approach" to semantic representation is any approach which invokes a particular logic in order to formalize the meaning. Propositional logic, predicate calculus, lambda calculus, etc… In what follows, I discuss Montague Grammar, as it combines elements of most other logicist approaches in some form (Boolean operators, predicates, quantifiers, variables, etc…). Montague Grammar is a theory of semantics set forth by the logician Richard Montague. As a system of formal semantics, Montague Grammar focuses on model-theoretic conception of semantics, truth-conditional aspects of meaning as well as the methodological centrality of the Principle of Compositionality. This principle states that "the meaning of a complex expression is a function of the meaning of its parts, and the syntactic rules by which they are combined." (Partee & al, 1993) An example of this principle in action could be something such as a parse tree (thus highlighting the important connection between the formalization of syntax and semantics):



As I stated, Montague Grammar states that the meaning of a sentences relies on its truth conditions. Therefore, a sentence such as "Mary sees the tree" is true if and only if Mary sees the tree. But remember the original goal: turn natural language into a formal language, so that it can be modeled computationally. So from this point, we can take the truth conditions, and represent them using logical formulae, such as:

Natural Language: Mary sees the tree.

Formal Language: $\exists x(\text{tree}(x) \wedge \text{see}(m^*, x))$

As we see from the formalization, the formal expression takes elements from propositional logic (truth conditions and Boolean operators), and first-order logic (predicates and quantification). The next step is to somehow implement this formal language expression in a computer environment. In order to do this, consider Chapter 10 of *Natural Language Processing with Python* entitled "Analyzing the Meaning of Sentences". In the text, the following example is used:

Natural Language: A dog disappeared.

Formal Language: $\exists x.(\text{dog}(x) \wedge \text{disappear}(x))$

NLTK Rendering: `exists x.(dog(x) & disappear(x))`

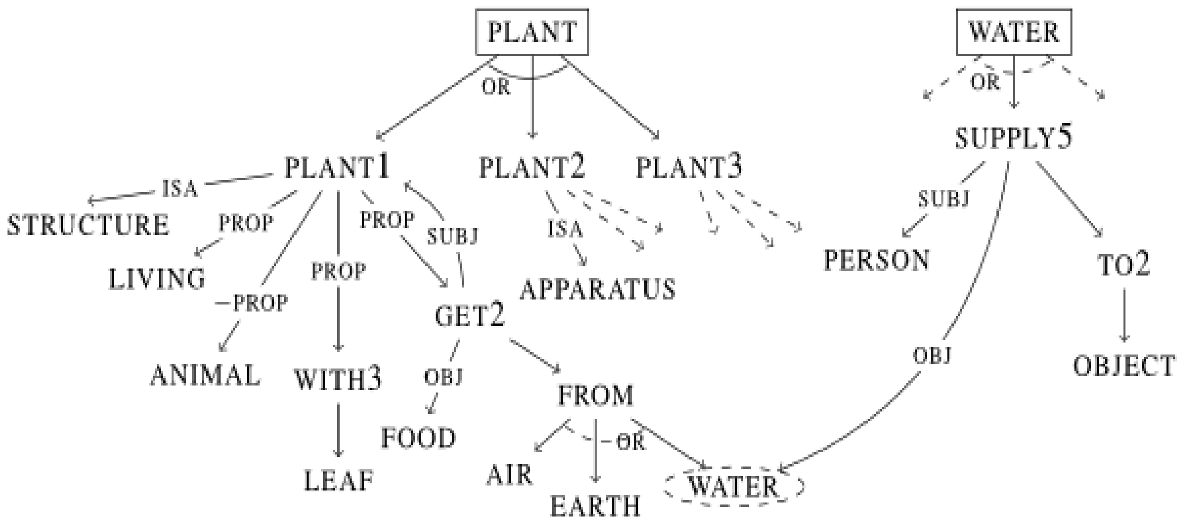
To further explain, the NLTK is the Natural Language Toolkit used by the object-oriented programming language Python in order to perform natural language processing and computational linguistics. Once the natural language has been converted to a logical formula, the NLTK can render the logic further so that the programming language can make sense of the formula. The NLTK can use a function known as `LogicParser()` in order to take in variables, and classify expressions. The NLTK rendering in the example above is acceptable by first-order logic standards (which is essentially encompassed within Montague Grammar), but Montague Grammar also implements elements of Type Theory into the model. This provides more thorough expressiveness and flexibility. Using type theory, we can create a set of basic types, such as {e, t}, where ‘e’ is the type of individual terms or “entity”, and ‘t’ is the truth value or type of formula. Even the NLTK implements type theory. This has only been a brief overview but the goal is not to necessarily discuss all the elements of Montague Grammar, but to show on a basic level how Montague Grammar uses formulas and methodologies of a variety of logicist approaches, and succeeds at the formalization and representation of semantics.

In summary, Montague Grammar:

- Combines elements of syntax and semantics in the formalization of natural language
- Uses elements of a variety of logics for semantic representation and modeling
- Implements elements of set theory and (more importantly) type theory
- Follows the Principle of Compositionality
- formalizes semantics so that we can perform natural language processing

2) Psychologically motivated approaches to meaning

Under this approach, there is less of an emphasis for representing semantics via the development of linguistic capabilities in machines, and more of an emphasis on modeling human cognition (or at least approaching meaning representation from this perspective). Approaches to representation under this view place emphasis on elements of the physical symbol system hypothesis, cognitive architectures, and conceptual models. The following model comes from the *Stanford Encyclopedia of Philosophy*:



This model is a graphical visualization of a way to enable **word sense disambiguation** through the act of "spreading activation". I'd argue that these approaches which draw their modeling techniques from cognitive modeling do so by making the following parallel:

$$\begin{tikzcd}\textbf{Cognitive (Mental) Models} \arrow[r] \arrow[d, dash] & \textbf{Linguistic Representation} \arrow[d, dash] \\ \textbf{Conceptual Models} \arrow[r] & \textbf{Semantic Representation}\end{tikzcd}$$

These "psychologically motivated" approaches base their ideas on connectionist and neural modeling in order to represent language (e.g. semantics) in a symbolic way. One problem with this approach is that it assumes for some of the proposed representations that mental processing can be **fully** and **accurately** described in terms of symbol manipulation. A (less radical) alternative to this view proposed by cognitive scientist Paul Smolensky is the **subsymbolic hypothesis.** As David Chalmers points out in a paper entitled "Subsymbolic Computation and the Chinese Room", the subsymbolic hypothesis says that connectionist models are better at working at the subsymbolic level, that is, a level below that of the symbol. As Chalmers states, "these models have no direct correspondence between the entities computed upon and entities in the world—but not necessarily as deep a level as that of the neuron. These systems still follow rules, but the rules are well below the semantic level. It is hoped that as a consequence of following rules at this low level, semantic properties will emerge—that is, manifest themselves in the processing and behavior of the program—without having been explicitly programmed in." (Chalmers, pg. 2)

I think that psychologically motivated approaches to meaning are the most difficult kind of semantic representations to implement. While I understand the approach is to understand and conceptualize semantics from a cognitive scientific perspective (after all, the mind is the source of semantics), I think that there is still a lot of development to be had in this approach in order to make it less assuming. Applying ideas such as the physical symbol system hypothesis as a symbolic way to talk about linguistics, specifically semantics, is still as of now a controversial idea – and this is why most often the methods employed for computational semantics usually draw from either formal semantics (as we see in the logicist approaches) or statistical semantics.

3) Statistical Semantics

Statistical semantics uses aspects of corpus linguistics and probability theory in order to extract the semantic properties of words in texts (corpora), and then categorize them based on their distributional characteristics. This has been a growing approach to performing natural language processing over the past decade, and continues to grow as a field within computational linguistics. In order to represent the meaning in statistical semantics, the sentences themselves are used (they are not necessarily/always formalized first). This means that instead of formalizing the natural language, as is the case with the logicist approaches, we instead use the sentences themselves (in conjunction with distributional knowledge) as the objects from which we can enable inferences. This is certainly a different assertion than the one we explored with Montague Grammar, but one aspect of Montague's approach to formal semantics remains intact: that language ***is*** logic. Montague believed that the reason we can formalize natural language at all is because language is just an expression of logic in a natural and human context/way. As Montague stated in his publication *Universal Grammar,* "There is in my opinion no important theoretical difference between natural languages and the artificial languages of logicians; indeed, I consider it possible to comprehend the syntax and semantics of both kinds of language within a single natural and mathematically precise theory."

Some, however, disagree with the extraction of semantic representations based on large text corpora, including Noam Chomsky. In a recent talk at Google, Chomsky expressed his opinion of statistical semantics, stating "[Getting significant knowledge from a language via the statistical analysis of texts] is extremely unlikely to succeed. You do not get discoveries in the sciences by taking huge amounts of data and throwing them into a computer and doing statistical analysis of them. Try to think it through in the history of the sciences; it just doesn't happen. That's not the way you understand things. You have to have theoretical insights; you have to know what kind of experiments to carry out, what data is worth looking at, and what data you throw away, and so on."

Limitations: Ambiguity, Context, and Common Sense

So far, I have brought forth and discussed three approaches to meaning representation in computational linguistics. However, even though I discussed the controversy and limitations of (2) Psychologically motivated approaches to meaning and (3) statistical semantics, I have yet to discuss the limitations of the logicist approaches, and this is because I think the limitations of these approaches apply to all of the approaches listed: ambiguity. In semantics, formal approaches often times rely on truth conditions and the meaning of the words themselves in order to turn the natural language into a formal language, however, there are aspects of meaning not captured by semantics. Semantics lacks the context of language, the social human aspect behind language, as well as the common sense knowledge needed to analyze text of a given dialogue. Take for example the following example of an ambiguous English sentence:

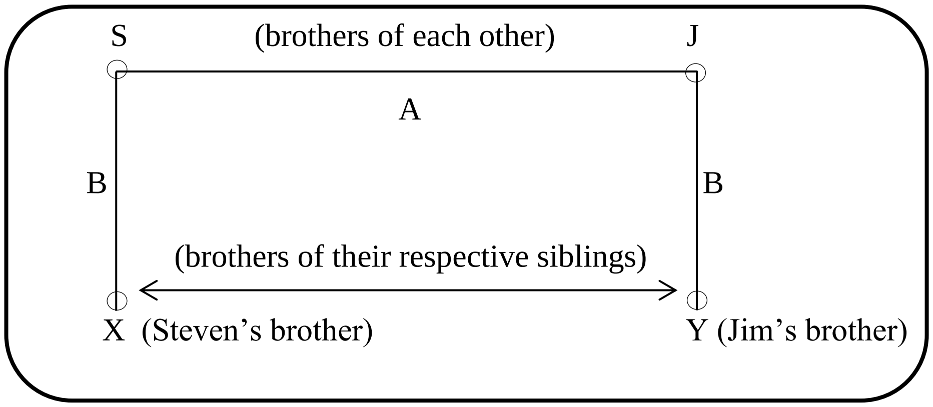
Sentence in English:	**You have a green light.**
Possible Meanings: **(showcasing the ambiguity)**	➢ It could mean that you have green ambient lighting. ➢ It could mean that you have a green light while driving your car. ➢ It could mean that you can go ahead with the project. ➢ It could mean that your body has a green glow. ➢ It could mean that you possess a light bulb that is tinted green.

Without knowing the context of the sentence, no amount of knowing the meanings of the words is going to tell you how this sentence is being used in the relevant dialogue. This is an example of ambiguity due to lack of context.

Another example of ambiguity in meaning is when we lack common sense knowledge of the social human world. Take for example the following sentence:

Steven and Jim are brothers.

This sentence seems simple enough, however the ambiguity comes from "are brothers". In this case, we seem to know, due to being English speakers and due to common sense that we mean to say that Steven and Jim are *the brothers of each other,* as in, they are siblings. However, this is not specified or addressed specifically in the text. Here is a model of this ambiguity:



As seen in this model, the line at 'A' represents the case that we mean that they are the brothers of each other, and the lines at B represent the chance that they are the brothers of their own respective siblings (indicated here as 'X' and 'Y'). In English, this kind of ambiguity is not problematic, and we don't even realize it. This is because we have a common sense understanding of the world around us, as well as a common sense understanding of how our language is used to represent that world. These are examples of pragmatic ambiguity, and for that we need the area of pragmatics.

II. Pragmatics

Pragmatics can be defined as the subfield of linguistics associated with the context of meaning. Although there are ways to make sense of text without invoking pragmatics, I'd argue that there are aspects of meaning ambiguity unobtainable to any other field of study within linguistics, and that pragmatics is therefore needed and relevant. In a lot of ways, at least with respect to meaning, pragmatics picks up where semantics leaves off.

In linguistics and the philosophy of language, one important consideration is **Speech Act Theory**. A speech act can be defined as an utterance with a performative function in communication. Examples include: promises, greetings, invitations, congratulations, bets, and orders. One fundamental aspect of speech acts is their apparent lack of truth conditions, a concept of which logicist approaches to formal semantics depended so heavily. But could any of the following sentences be considered "true" or "false"?:

- Go turn off the TV!
- I bet I can run to the top of that mountain!
- Congratulations!

Thanks to philosopher J.L Austin, speech acts (performative utterances) are typically broken down in the following way:

Locutionary act: the performance of the utterance itself; the phonetic aspect

Illocutionary act: the meaning behind the utterance, the social significance and force

Perlocutionary act: the effect of the utterance in the social context (such as persuasion)

A successful system of computational pragmatics, therefore, should be any system that can formalize the aspects of the social context so that it can be understood in a computational environment. At the current time, it may be worth noting that computational pragmatics as a subfield of computational linguistics is not as advanced as computational semantics. That does not mean, however, that there are no models for pragmatics in computational linguistics. The solution to pragmatic problems in computational linguistics heavily relies upon the problem itself, whether it be dealing with speech acts, reference resolution, or interpreting the context in the form of discourse analysis. In what follows, I discuss two computational approaches to pragmatics.

The Plan Inference (or BDI)

In this approach, often called the BDI (belief, desire, and intention) or PLAN-BASED model, the approach seeks to interpret speech acts. We can apply the concepts of thematic roles (such as agenthood) to the speaker(s) or "dialogue participants" in order to assign unto that agent not only the speech acts themselves, but the common sense knowledge behind those speech acts which are being used to progress the dialogue. In a BDI software architecture, you can therefore go beyond the actual words being said in order to comprehend the context behind them. The BDI system breaks down as follows:

Beliefs	Beliefs represent the informational state of the agent (this can include inference rules which allows forward chaining to lead to new beliefs in the dialogue). Beliefs such as these are stored in the "belief set" database within the architecture of the BDI system.
Desires	Desires are the motivational state of the agent. That is, the goals and objectives of the agent in the dialogue.
Intentions	The deliberate state of the agent. The intentions reflect the decisions that the agent has made based on the speech acts, and recognize these decisions as "plans" or deliberate sequences of actions.

As we can see, we are capturing much more about the dialogue/speech acts themselves with this model than we ever could by just breaking down words with formal semantics. We are only now truly beginning to get at the intentions of the speaker and the social context. However, this model has its criticisms. For example, this model does work by creating a dialogue related context around an intelligent agent, but not necessarily multiple agents. Therefore, this idea of agenthood that the model proposes is not clearly a part of a multi-agent system. Also, at least at the current time, BDI models seem to lack any capability of machine learning, that is, there is no mechanism in the architecture for "learning" from past behaviors in order to adapt to new situations overtime.

DAMSL (Dialogue Act Markup in Several Layers)

DAMSL is a system that is better at working with multiple agents (the limitation in this case being we can only use two agents). This scheme is specifically developed to capture the "dialogue acts" of two agents in a task-oriented dialogue, wherein which two agents are participants of collaboration, geared towards solving a given problem. DAMSL works as a tag set, tagging utterances in order to record the agent's purpose and role in the dialogue, as well as the specific purpose and role of the utterance itself. Perhaps the most important and prominent feature of DAMSL is how the tag set distinguishes between the forward looking and the backward looking function of an utterance.

Function Type:	**Primary Operation of the Function:**
Forward Looking	Recognizes that the progression of dialogue can change the beliefs, positions, and future actions of the agents, and therefore annotators are allowed to "look ahead" in the dialogue to determine the effect an utterance has/had on the dialogue (Includes statements, info-request, exclamations, open-options, offers, etc…).
Backward Looking	Recognizes how current utterances relate to previous ones throughout the discourse. Considers the relationship between utterances such as responses and answers to the previous (backward) utterances (Includes agreements, acceptance, demonstrations of comprehension, answers, acknowledgement, etc…).

One important question to consider is: do either of these approaches to understanding and implementing dialogue contribute to the dispersion of the meaning ambiguity? I would argue that they do. In natural language processing, the problem of word-sense disambiguation primarily occurs because the dialogue, context, and common sense knowledge that comes from knowing about the dialogue and context is not specifically addressed in the formal semantics. Certainly computational approaches to pragmatics are not the only way to combat this problem (machine learning techniques and knowledge based methods also try to combat this problem). However, since pragmatics is the area of linguistics primarily concerned with the context of the dialogue (as seen from the approaches above) then these ambiguities due to a *lack* of context are best handled by implementing a system of pragmatic study on a computational level. As of now, computational pragmatics still needs major development, but once the subfield grows, this will improve not only our understanding of pragmatics, but of semantics, computational linguistics, and natural language processing.

In this paper, we looked at semantics and pragmatics in computational linguistics. First we looked at the approaches to representing semantics on a computational level: logicist approaches, psychologically motivated approaches (representationalist and connectivist), and statistical semantics. Then, after discussing the limitations of computational semantics due to ambiguity from lack of social context and common sense knowledge, we moved on to approaches to computational pragmatics, and how they actually implement systems which recognize the dialogue, and draws from Speech Act Theory in order to formalize utterances (such as tagging them in a task-oriented scheme). Therefore, the need for both semantics **and pragmatics** is crucial for the most accurate and detailed form of meaning representation in natural language processing and computational linguistics.

References

"Noam Chomsky | Talks at Google." *YouTube*. Talks at Google, 8 Apr. 2014. Web. 15 May 2014. <http://www.youtube.com/watch?v=Y3PwG4UoJ0Y>.

“Pragmatics” *Wikipedia, The free Encyclopedia.* Wikimedia Foundation. Web. Retrieved 15 May 2014. < http://en.wikipedia.org/wiki/Pragmatics >

Austin, J.L. “How to Do Things With Words” Cambridge. Harvard University Press. 1962. 2nd Edition, 2005. Print.

Bird, Steven, and Ewan Klein. *Natural language processing with Python*. Beijing: O'Reilly, 2009. Print.

Chalmers, David. “Subsymbolic Computation and the Chinese Room” *Indiana University*. Web. Retrieved 15 May 2014. <http://consc.net/papers/subsymbolic.pdf>

Core, Mark, and James Allen. "Draft of DAMSL: Dialog Act Markup in Several Layers." Rochester University, n.d. Web. 15 May 2014.

Phung, Toan and Winikoff, Michael. “Learning Within the BDI Framework: An Empirical Analysis.” *Knowledge-Based Intelligent Information and Engineering Systems.* 2005. Retrieved 15 May 2014. <http://link.springer.com/chapter/10.1007%2F11553939_41>

Schubert, Lenhart. "Computational Linguistics." Stanford University. Stanford University, 6 Feb. 2014. Web. 14 May 2014. <http://plato.stanford.edu/entries/computational-linguistics/>.

Thater, Stefan. “Montague Grammar” Web. 15 May 2014. <http://www.coli.uni-saarland.de/courses/underspecification-06/slides/01-mg.pdf>