Linguists and philosophers have tried to classify all sentences according to their power to do and influence the world (the speech act). Unfortunately, conventional use frequently allows for the speech act to misalign with the syntax. This indirect speech act has the illocutionary force of the equivalent direct speech act. One theory of disambiguating direct and indirect speech acts includes preconditions which also must be true for an act to occur (felicity conditions). For instance, the QUESTION speech act requires the addressee to:

- know the answer
- have the ability to answer the question, if posed

I have used CASSIE, an intensional rational agent who can reason about the knowledge of others, as an environment for categorizing indirect speech acts by way of the felicity conditions.

CASSIE does not have a complete natural language processor. In order to understand the three sentences below, the speaker's intent would have to be specifically coded. This causes problems with scalability and polysemous questions (e.g., the middle example can be interpreted as a request or question).

<table>
<thead>
<tr>
<th>Can you tell me the time?</th>
<th>Can you tell me the time?</th>
<th>Can you tell me the time?</th>
</tr>
</thead>
<tbody>
<tr>
<td>less literal</td>
<td>Most Likely Interpretation</td>
<td>more literal</td>
</tr>
</tbody>
</table>

In fact, most natural language systems suffer from the same ambiguity issues. Even with complete morphosyntactic and semantic knowledge, a parser needs some heuristic to resolve this problem. I explicitly deal with SneRE, the rational engine module of SNePS. It provides the system with the ability to act and plan according to knowledge stored in and inferences made on the network. Above, m2! is the proposition that the act "tell" has the plan represented by proposition m3.

The ! (bang) labels those nodes which are believed by the system (for our purposes, CASSIE). This contrast can help us to differentiate between CASSIE knowing an Act-Plan and the abstract concept of that Act-Plan.

The Final Implementation

The first modification is to add an extra level of abstraction attached to each sentential proposition. In this example, m1 represents the information encoded by the utterance and m2! represents a particular instantiation of m1. For now, it includes the syntax and illocutionary force of m1. In the future, it could include features such as focus or language (e.g., Russian).

Finally, we make sure to seed CASSIE with sufficient knowledge about which acts she can perform and who knows she can perform them. She now has the semantic/syntactic structure (step 1), the reasoning ability (step 2), and belief knowledge to properly apply the felicity conditions.

CASSIE and SNePS

CASSIE is a computational agent implemented in SNePS, a LISP-based network of nodes and arcs. Each node represents some concept or proposition and each arc is a labeled relation.

For instance, the proposition "colorless ideas" can be depicted by the node m1! with an object arc to word "idea" and a property arc to the word "colorless."

I have used CASSIE, an intensional rational agent who can reason about the knowledge of others, as an environment for categorizing indirect speech acts by way of the felicity conditions.

Prior Computational Work

- The indirect speech act is inferred from a failure of the direct act
- Computational models use chains of belief and inference to find an appropriate interpretation
- The interlocutors’ intentions and beliefs are important.
- Most evidence is drawn from written language.

**Problems**

- Levinson (1983) notes most speech acts are indirect.
- Swinney and Cutler (1979) show parallel access of direct and indirect meaning.
- Is it really safe to say that all statements have a single intent and meaning?

- The surface form has encoded cues to direct the decoding by the addressee.
- Probabilistic modeling and learning techniques are used to classify speech acts.
- Most evidence is drawn from spoken language.

**Problems**

- Of course, to associate the cues with tags, any system must be trained on a hand-coded corpus.
- Most explanatory value is obfuscated by a Hidden Markov Model.

References


Future Work

- Investigate various tag sets and implement more elaborate acts
- Compare automated results with human-tagged utterances
- Create a training set for learning felicity conditions