Computational Semantics and Pragmatics

Autumn 2011

Raquel Fernández

Institute for Logic, Language & Computation
University of Amsterdam

Last Week

Gricean pragmatics: conversational principles (the maxims) can be exploited to derive inferences (implicatures) that enrich the literal meaning of an utterance.

Generation of Referring Expressions: how to generate expressions in accord with Gricean principles, in particular that contain *the right amount of information*.

- Algorithms implementing different interpretations of the Maxim of Quantity/Brevity.
- Incremental Algorithm: in theory closer to human expressions, computationally less costly than stricter interpretations of the maxims.
- Evaluation of GRE algorithms relies on heavily annotated corpora, such as the Tuna corpus from HW#3.

The Incremental Algorithm - simplified

Let:

- r be the target referent;
- A be the set of properties $a{=}v$ that characterise r;
- C be the set of distractors (the contrast set);
- $\texttt{RulesOut}(a{=}v)$ be the subset of C ruled out by property $a{=}v \in A$;
- P be an ordered list of task-dependent *preferred attributes*; and
- L be the set of properties to be realised in our description.

$\texttt{MakeReferringExpression}(r, C, P)$

$L \leftarrow \{\}$
for each member a_i of list P **do**
    if $\texttt{RulesOut}(a_i{=}v) \neq \emptyset$ (for some $a_i{=}v \in A$)
    then $L \leftarrow L \cup \{a_i{=}v\}$
        $C \leftarrow C - \texttt{RulesOut}(a_i{=}v)$
    endif
    if $C = \{\}$ **then**
        if $\{\texttt{type=}v\} \in L$ (for some value v such that $\texttt{type=}v \in A$)
        then return L
        else return $L \cup \{\texttt{type=}v\}$
        endif
    endif
return failure

Plan for Today

Computational, data-driven approach to resolving indirect answers to polar questions (de Marneffe, Manning & Potts, 2010)

de Marneffe, Manning & Potts (2010) "Was it good? It was provocative." Learning the meaning of scalar adjectives. In *Proceedings of ACL 2010*, pp. 167-176.

(1) A: Is it tasty? B: It is edible.	(2) A: Are you coming to the party? B: I have work to do.

The authors focus on indirect answers such as (1), where so-called scalar implicatures may be at play.

Scalar Implicature

The term *scalar implicature* refers to conversational implicatures that are related to the Maxim of Quantity and that can be characterised in terms of *scales*.

- Maxim of Quantity:
  - Make your contribution as informative as is required (for the current purposes of the exchange).
  - Do not make your contribution more informative than is required.

Scales

Scales are orderings of contrastive alternatives that differ in their semantic/pragmatic *strength*. There are two main types of scales:

- Entailing scales (Horn-scales): stronger expressions within the scale entail weaker expressions but not vice versa.

`<all, some>`	All boxes exploded $\models$ Some boxes exploded
`<excellent, good>`	The movie was excellent $\models$ The movie was good
`<...,ten, nine, eight,...>`	I have ten euros $\models$ I have nine euros

- Non-entailing scales: stronger expressions within the scale are informationally richer than weaker ones, but do not entail them.

`<succeed, try>`	John succeeded without even trying
`<general, colonel, lieutenant>`	John is a general $\not\models$ John is a colonel

Scalar Implicature

Given a scale <S,W> with strong (S) and weak (W) expressions, the use of W implicates the negation of S.

<all, some>	All boxes exploded $\models$ Some boxes exploded Some boxes exploded $\leadsto$ Not all boxes exploded
<excellent, good>	The movie was excellent $\models$ The movie was good The movie was good $\leadsto$ It was not excellent
<...,ten, nine, eight,...>	I have ten euros $\models$ I have nine euros I have nine euros $\leadsto$ I don't have ten euros
<succeed, try>	John succeeded without even trying John tried to set up a studio $\leadsto$ John didn't succeed
<general, colonel>	John is a general $\not\models$ John is a colonel John is a colonel $\leadsto$ John is not a general

Negation reverses the scales: the use of $\neg$S implicates W.

<all, some>	Not all boxes exploded $\leadsto$ Some boxes did.
<...,ten, nine, eight,...>	I don't have ten euros $\leadsto$ I have less than ten.
<succeed, try>	John didn't succeed to set up a studio $\leadsto$ He tried.

de Marneffe, Manning & Potts (2010) "Was it good? It was provocative"

- Phenomenon studied: indirect answers to polar questions where a "yes" / "no" answer is not explicitly given.
- Focus on two types of indirect question-answer pairs (QA):
  - QA1: Both the answer and the question include a gradable modifier:

A: Do you think this is a good idea? B: I think it is an excellent idea.	A: Is he qualified? B: I think he's young.

  - QA2: The question includes a gradable modifier and the answer a numerical expression:

A: Are your kids little? B: I have a 7-year-old and a 10-year-old.	A: Have you been living here very long? B: I've been here about 12.5 years.

Gradable Adjectives

- Gradable adjectives refer to properties that are gradable (that may hold to different degrees).
  - ∗ They admit adverbs of degree such as *very*, *rather*, *highly*, *fairly*, *slightly* and comparative/superlative morphology.
  - ∗ Most adjectives are gradable (e.g. *interesting, unusual, tall, little, expensive*), but not all are (e.g. *vegetarian, mammal, impossible*).
- They are also called *scalar adjectives* because the different degrees to which the relevant property may hold form a scale:

  `<excellent, very good, rather good, good>`

Gradable Adjectives

Most gradable adjectives have the following two features:

- Context-dependence: What counts as tall or expensive depends on the class of entities being considered. For instance, 3 Euros may count as expensive if we are talking about cups of coffee, but not so if we are talking about sandwiches.
- Vagueness: Even when we know the class of entities we are talking about, there isn't a sharp line delimiting what counts as not expensive, expensive, or very expensive – there are always borderline cases.

Grounding the Meaning of Scalar Adjectives

What do we need to resolve these two types of indirect QA pairs – to infer yes/no from the indirect answer?

- QA1: Both the answer and the question include a gradable modifier:

A: Do you think this is a good idea? B: I think it is an excellent idea.	A: Is he qualified? B: I think he's young.

⇒ Need to find out the relevant scale and relative ordering of the adjectives in the question and the answer.

- QA2: The question includes a gradable modifier and the answer a numerical expression:

A: Are your kids little? B: I have a 7-year-old and a 10-year-old.	A: Have you been living here very long? B: I've been here about 12.5 years.

⇒ Need to find out whether the numerical expression counts as a positive or negative instance of the adjective.

Corpus

The authors develop methods to deal with QA1 and QA2. To evaluate them, they need a corpus with some sort of gold standard annotation.

- Corpus of 205 QA1 pairs and 19 QA2 pairs from CNN interviews and the Switchboard corpus of telephone conversations.
- Use of Mechanical Turk to annotated each pair with an *answer assessment*: yes, probably yes, probably no, no, uncertain.
- Inter-annotator agreement is assessed with a version of kappa and with *entropy*.
- Uncertainty is important because not all indirect answers are intended to convey a clear yes / no – the speaker may not be committed to a particular answer.

QA1: Learning Scales

Knowing the relative scalar ordering of the modifier in the question P_Q and in the answer P_A gives us a way to infer the conveyed answer. Assuming "$>$" stands for "stronger than", then intuitively:

$P_A > P_Q$ → yes
$P_Q > P_A$ → no
$P_Q \ ? \ P_A$ → uncertain

- To derive the strength of an adjective, a corpus of online movie reviews is used:
  - ∗ The strength of an adjective w depends on the rating category (the number of stars) most commonly associated with it.
  - ∗ Expected rating value ER(w): average rating category for w
- A simple algorithm is then used to decide what answer is conveyed on the basis of the derived strengths or ERs.

QA2: Interpreting Numerical Answers

For each adjective and modified type of entities, they search the Web for positive and negative instances.

What counts and does not count as "little kids"?

``little kinds'', $n$ ``year-old'' $\rightsquigarrow$ n = positive instance
``not little kinds'', $n$ ``year-old'' $\rightsquigarrow$ n = negative instance

- The gathered numerical expressions automatically classified as pos / neg are used to calculate the probability that a particular n is a positive of negative instance of the relevant class.
- The probabilities are categorised as follows to decide what answer is conveyed:

$0 - 0.33$	$\longrightarrow$ no
$0.33 - 0.66$	$\longrightarrow$ uncertain
$0.66 - 1$	$\longrightarrow$ yes

Evaluation

- To evaluate the decision procedures for QA1 and QA2, their output needs to be compared to the gold standard given by the Mechanical Turk annotations.
- This requires some manual processing/annotation of the corpus:
  - The QA1 algorithm requires identifying P_Q and P_A and detecting the presence of negation
  - The QA2 procedure requires identifying the adjective ("little"), the modified entity ("kids") and the unit of measure ("years old") to be able to construct the Web queries.
- Measures: accuracy of each algorithm with respect to the gold standard and Recall, Precision, and F1 for each response type.
- The accuracy of the system correlates with the level of agreement amongst annotators.

Summary

- Data-driven approach: grounding modifier scales and pos/neg instance of gradable adjectives on data.
- Rule-based decision procedures for assigning the conveyed polarity (yes/no) of the indirect answer.
- A probabilistic approach could also be possible using supervised machine learning on the annotated corpus.

What's Next

- Three sessions left + presentations of final project
- We'll look into issues related to Dialogue Modelling
  - ∗ to get an overview, read my draft chapter on Dialogue
- Spend time working on your final projects!!