Modeling Information Structure & Generating Prosody

-Amritaansh Verma
What is Prosody?

- Prosody is the rhythm, stress and intonation (pitch) of speech.
- It reflects the emotional state of the speaker, form of the utterance (statement, question or command).
- Presence of sarcasm, emphasis, contrast and focus.
- Conveys hidden information not encoded by grammar or choice of vocabulary, example: yeah that’s great (sarcastically or happily makes a BIG difference).
Why is it important?

- Makes dialogue systems more life-like (think of Siri or an airline IVRS understanding your emotions!)
- Allows us to extract hidden/subtle information which is otherwise discarded
- This information enables us to understand speech in a better, more complete way and respond appropriately
What are we trying to achieve?

- Given transcripts of dialogues (also called corpus) we are trying to analyze and annotate them.
- We want to describe its information structure, organization and salience.
What is Information Structure?

- Information structure describes the **availability**, **organization** and **salience** (also called kontrast) of entities in a discourse.

- **Availability** refers to the information status (*old, mediated, new*), it expresses the **availability** of entities in discourse.

- **Organization** refers to *theme/rheme* of an intonation phrase.
What is Information Structure?

- **Saliency/kontrast**: how salient (Most noticeable or important) the speaker wishes to make an entity

- Example: “You need to pull the door NOT push it!”
Switchboard Corpus

- Consists of 2430 spontaneous phone conversations
- Average duration: 6 minutes
- Speakers use American English
- A Total of three million words
- A subset of the corpus was used for annotation
What’s a Switchboard?
Information Status/Availability

- Information Status describes how available an entity is in the conversation.
- Defined in terms of the speaker’s assumptions about the hearer’s knowledge/beliefs.
- Expressed by the well-known old/new/mediated distinction.
Some ground terms used in Information Structure

- **NEW**: An entity which has not been previously referred to and is yet unknown to the hearer.

- **MEDIATED**: An entity that is newly mentioned in the dialogue but that the hearer can infer from prior context or universal knowledge.

- **OLD**: An entity is old when it is not new nor mediated.
Two more Categories

- Non-Applicable: Wrongly extracted markables, like “course” in “of course”
- Not-Understood: Used when the annotators find some entities too difficult to understand
- Entities marked as non-applicable or not-understood are excluded from further annotation
- All other entities must be classified into old, mediated, or new.
NEW

- The category new is assigned to entities that have not yet been introduced in the dialogue and that the hearer cannot infer from previously mentioned entities.

- No subtypes are specified for this category.

- Example: “Hey, did you eat at the PizzaStudio on Figueroa street?”
Mediated

- Mediated entities are inferable from previously mentioned ones, or generally known to the hearer.
- Nine subtypes: general, bound, part, situation, event, set, poss, func value, aggregation
Quick examples for subtypes of Mediated

- General - Generally known entities such as “the moon” or “Italy” or “The pope” or “The Spiderman”
- Bound - Pronouns, such as “them” in “it’s hard to raise one child without them”
- Poss - mark all kinds of intra-phrasal possessive relations
- (part, situation, event, and set) - used to mark instances of bridging
Quick examples for subtypes of Mediated

- **Part:** Used to mark part-whole relations, Example:

  "When I come home in the evenings my dog greets me at the door."  
  "The door" is a part of the whole house

- **Event** – Applied to an entity related to a previously mentioned verb phrase, Example:

  "We were travelling around LA, and the bus was really full."  
  "the bus" is triggered by travelling around LA.

- **Aggregation:** Used for coordinated noun phrases.
OLD

- An entity is old when it is not new nor mediated.
- This is usually the case if an entity is coreferential with an already introduced entity.
Hierarchical Scheme

- Finer-grained distinctions have proved hard to distinguish reliably in practice, therefore we organized our schema hierarchically.

- We use the three main classes described above as top level categories for which more specific subtypes can be assigned.

- Preserves a high-level, more reliable distinction while allowing a finer-grained classification to be exploited for specific tasks.
How good was this scheme?

- Three Switchboard dialogues (for a total of 1738 markables) were marked up by two different annotators.
- Evaluated annotation reliability by using the Kappa statistic.
- Good quality annotation of discourse phenomena normally yields a kappa (K) of about .80.
- This scheme had K = .845 for high-level categories and K = .788 when including subtypes.
- Later applied for the annotation of a total of 147 Switchboard dialogues.
In layman’s terms

- Information status describes how available an entity is in a discourse.
- Generally *old* entities are available, and *new* entities are not.
- Mediated entities fall mid-way between new and old but closer towards old.
However this is only one aspect of the information...

- We also need to describe how speakers signal the organization and salience of elements in discourse.
- For this there are two notations: theme/rheme and background/kontrast.
Theme/Rheme

- Theme/rheme structure guides how an element fits into the discourse model
- If it relates back it is *thematic*
- If it advances the discourse it is *rhematic*

*Example:* "who is he?" "he is a student"

The "he is" part of the second clause is the clause's theme and "student" is the rheme
Background/Kontrast

- Kontrast relates to Salience i.e emphasis
- We expect new entities to be salient and old entities not
- Therefore, if an old element is salient, or a new one especially salient, an extra meaning is implied.
Example of theme/rheme and background/kontrast

(Q) Personally, I love hyacinths.

What kind of flowers grow in your area?

(A) (In MY area) (Theme)

(it is the DAFFODIL) (Rheme)

Bkgd Kont. Bkgd Bkgd Kont.
A more subtle theme/rheme example

- We define the rheme as any prosodic phrase that is not identifiable as a theme.

- Annotators mark each prosodic phrase as a theme if it only contains information which links the utterance to the preceding context.

“I lived in England for four years”

“Where I Lived” (Theme)

“was a town called Newmarket” (Rheme)
Although there is a clear link between prosodic prominence and kontrast, there are a number of disagreements about how this works.

Some, claimed that kontrast within theme and kontrast within rheme are marked by categorically distinct pitch accents.

Another view is that kontrast only applies to themes that are contrastive; the head of a rheme phrase always attracts a pitch accent, it is therefore redundant to call one part kontrastive.

Further, some consider kontrast within a rheme phrase only occurs when there is a clear alternative set, Example: *daffodil* contrasts with other flowers the speaker might grow.
How to solve this?

- Rather than using the abstract notion of kontrast directly, we will identify discourse scenarios which commonly invoke kontrast.

- Basically, in simple terms, using the full discourse context including the speech, annotators mark each content word for the first category that applies. If none apply, they mark it as background.
Categories of Kontrast

- **Correction**: The speaker’s intent is to correct or clarify another just used by them or the other speaker, example: the speaker wishes to clarify whether her interlocutor really meant “hyacinths”. (now are you sure they’re HYACINTHS) (because they do not grow in this area)

- **Contrastive**: The speaker intends to contrast the word with a previous one, example:

  I wanted RED apples, but all they had were GREEN
Categories of Kontrast

- **Subset**: The speaker highlights one member of a more general set, Example, the speaker introduces “three iPhones”, and then gives a fact about each.

  (THIS girl owns *THREE iPhones*) *(TWO for work) (and ONE for home) (and she had to buy the SECOND one for work) (because her battery didn’t last all day)*

- **Adverbial**: The speaker uses a focus-sensitive adverb

  Example: B didn’t even like the “trailers” of ‘G I Joe’, let alone the movie.

  (A) I like Dwane Johnson, though I thought he was crummy in ‘G I Joe’.  
  (B) (I didn’t even like) (the TRAILERS)
Categories of Kontrast

- **Answer:** The word and no other, fills to an open proposition set up in the context, Example: A sets up the “blooms” she can’t identify, and B answers “lily”.

(A) We have *these blooms*, I’m not sure what they are but they come in all different colors yellow, purple, white...

(B) *(I BET you) (that that’s a LILY)*
Something more Exciting!

http://www.sitepal.com/ttswidgetdemo/


**BEAT Example**: Reaction to a speech which is boring (Not exactly we are discussing)
BEAT

• BEAT Stands for the Behavior Expression Animation Toolkit

• Input: Allows animators to input typed text

• Output is appropriate and synchronized nonverbal behaviors (Hand gestures, facial expressions) and synthesized speech by a animated human figure

• The nonverbal behaviors are assigned on the basis of actual linguistic and contextual analysis of the typed text, using rules derived from extensive research into human conversational behavior
BEAT

- The toolkit is extensible, so that new rules can be quickly added.

- It is designed to plug into larger systems that may also assign personality profiles, motion characteristics, scene constraints, or the animation styles of particular animators.
How is BEAT different?

• Typical procedural animation systems are capable of generating extremely realistic movement, hand gestures, and facial expressions in silent characters.

• However, when voice is also added, the issues of synchronization and appropriateness crop up.

• But we still want to animate animated characters which speak...
A typical error/disfluency

- Untrained animators and autonomous animated interfaces, often generate a pointing gesture towards the listener when a speaking character says “you”, example: “If you want to come with me, get your coat on”

- However this never occurs in life (rarely if there is some ambiguity about whom ‘you’ refers to)

- Makes the animated character appear unnatural as if speaking a language not her own

- In some cases, we may not even have the opportunity to handcraft or capture the animation. example: interfaces to web content, animated non-player characters in interactive role playing games, and animated avatars in online chat environments
What is typically done?

- Many animators rely on video footage of real actors reciting the text, for reference or rotoscoping, or more recently, rely on motion captured data to drive speaking characters.

- This is expensive and tedious!

- May be worth doing for characters that play a central role on the screen, but is not as justified for a crowd of extras.
How is BEAT better?

- BEAT allows one to animate a human-like body using just text as input.
- Cheaper, easier and more reliable.
- The mapping from text to speech+ (facial, intonational and body) gestures is contained in a set of rules derived from the state of the art in nonverbal conversational behavior research.
- Importantly, the system is extremely permeable, allowing animators to insert rules of their own concerning personality, movement characteristics, and other features that are realized in the final animation.
BEAT System Architecture

**Figure 1. BEAT System Architecture**
BEAT Components

- BEAT components are written in JAVA, and thus are easily extensible.

- Each BEAT component is basically an XML Transducer, i.e. it takes in tagged text as input and produces tagged text as output.

- Initial input is the script text entered by the user.

- Each module operates by reading in XML-tagged text, converting it into a parse tree, manipulating the tree, then re-serializing the tree into XML before passing it to the next module.
Knowledge Base

- It adds some basic knowledge about the world to what we can understand from the text itself
- Allows us to draw inferences from the typed text
- Allows us to specify the kinds of gestures and the kinds of places where emphasis should be placed
- Currently the KB is stored in two XML files
- Object.xml and Action.xml
Object XML file

- The object knowledge base contains definitions of classes and instances of objects.

- The sample defines a new object class CHARACTER as a type of person with two features: TYPE, describing whether the professional is REAL or VIRTUAL; and ROLE, describing the actual profession.

- Each feature value is also described as being "normal" or "unusual" (e.g., a virtual person would be considered unusual), which is important since people tend to generate iconic gestures for the unusual aspects of objects they describe.
<CLASS NAME="CHARACTER" ISA="PERSON">

<FEATURE NAME="TYPE">
  <VALUEDESC NAME="REAL" ISNORMAL="TRUE">
  </VALUEDESC>
  <VALUEDESC NAME="VIRTUAL" ISNORMAL="FALSE" GESTURE="gesture specification goes here">
  </VALUEDESC>
</FEATURE>

<FEATURE NAME="ROLE">
  <VALUEDESC NAME="ACTOR" ISNORMAL="TRUE">
  </VALUEDESC>
  <VALUEDESC NAME="ANIMATOR" ISNORMAL="TRUE">
  </VALUEDESC>
</FEATURE>

<INSTANCE NAME="PUNK1">
  <VALUE FEATURE="ROLE" VALUE="ACTOR">
  </VALUE>
  <VALUE FEATURE="TYPE" VALUE="VIRTUAL">
  </VALUE>
</INSTANCE>

</CLASS>

Figure 3. Example Object Knowledge Base
Action XML File

<ACTION NAME="MOVE" GESTURE="R hand=5, moves from CC towards L ...">

- The action.xml contains associations between domain actions and hand gestures which can depict them.
What else does BEAT include?

- The system comes loaded with a generic knowledge base, containing information about some objects and actions, and some common gestures.
- Gestures are specified using a compositional notation in which hand shapes and arm trajectories for each arm are specified independently.
- This makes the addition of new gestures easier, since existing trajectories or hand shapes can be re-used.
Language Tagging Module

- The largest unit is the UTTERANCE (Entire paragraph)
- UTTERANCE -> CLAUSE (A PROPOSITION)
- CLAUSE -> THEME + RHEME
- THEME: Part of the clause that creates a coherent link with a preceding clause
- RHEME: Part that contributes some new information to the discussion
- Example: "who is he?" "he is a student"

the "he is" part of the second clause is that clause's theme and "student" is the rheme
Language Tagging Module

- (THEME+RHEME)-> word phrase
- Word phrase, (2\textsuperscript{nd} smallest unit) describes an ACTION or an OBJECT.
- ACTION: verb phrase
- OBJECT: noun phrase
- If an exact match for a verb is not found, it is sent to a word ontology module which creates a set of hypernyms, like “moving” is a hypernym for “walking”, then moving is searched in the knowledge base

A hypernym of a word is a related, but a more generic or broader term.
Language Tagging Module

- The smallest units that the language module handles are the words themselves.
- Tagger uses the EngLite parser from Conexor (www.conexor.fi) to supply word categories and lemmas for each word.
- Keeps track of all previously mentioned words and marks each incoming word as NEW if it has not been seen before.
- This “word newness” helps to determine which words should be emphasized by the addition of intonation, eyebrow motion, or hand gesture.
- Also tags contrasting words like “red” and “green”.
- Each word in a contrast pair is tagged with the CONTRAST tag.
Language Tagging Module

Summary

- Clause
- Theme and Rheme
- Word newness
- Contrast
- Objects and actions
Flow Chart: 1

UTTERANCE

It is some kind of a virtual actor.

a. Input to Language Tagging Module
Flow Chart: 2

b. Output from Tagging Module / Input to Generation
Flow Chart: 3

c. Output from Generation Module / Input to Scheduling Module
d. Output from Scheduling Module (flattened tree)
Behavior Suggestion

- Operates on the XML trees produced by language tagging module
- Augments them with suggestions for appropriate non-verbal behavior
- Applies a set of behavior generators to all nodes in the XML tree
Behavior Generators Implemented so far

- Beat GestureGenerator (idling hand motion)
- Surprising Feature Iconic Gesture Generator
- Action Iconic Gesture Generator
- Contrast Gesture Generator
- Eyebrow Flash Generator
- Gaze Generator
- Intonation Generator
Behavior Selection

- Behavior Selection module analyzes the tree that contains many, potentially incompatible, gesture suggestions.
- Reduces these suggestions down to the set that will actually be used in the animation.
- The selection process utilizes a set of filters which are applied to the tree in turn, each of which can delete behavior suggestions which do not meet its criteria.
- Filters reflect the personalities, affective state and energy level of characters by regulating how much nonverbal behavior they exhibit.
- Currently, two filter strategies are implemented: conflict resolution and priority threshold.
Conflict Resolution Filter

- Conflict resolution filter detects all nonverbal behavior suggestion conflicts i.e. those which physically cannot co-occur,
- Like winking and rolling eyes!
- Resolves the conflicts by deleting the suggestions with lower priorities
Priority Threshold Filter

- The priority threshold filter simply removes all behavior suggestions whose priority falls below a user-specified threshold.
Behavior Scheduling and Animation Module

- Converts its input tree into a set of instructions which can be executed by an animation system.

- Two ways to achieve synchronization between a character animation subsystem and a subsystem for producing the character's speech:
  - The first is to obtain estimates of word timings and construct an animation schedule prior to execution.
  - The second approach is process real-time events from a TTS engine, generated while the TTS is actually producing audio, and compile a set of event-triggered rules to govern the generation of the nonverbal behavior.

- Both approaches were used in BEAT.
Figure 7. Scheduling Process
BEAT in action
Thank You!