Dialog State Tracking Based on Pairwise Ranking

Veljko Miljanic

Dialog State Tracking

- State is representation of what the user wants at any point
    - Slot values, requested slots, method
- Accumulate evidence over the sequence of dialog
    - SLU hypothesis, ASR, previous system actions, …
- Improves robustness of the system
    - ASR errors
    - SLU errors

DSTC2 Dataset

- Dialog State Tracking Challenge
  - Dialog corpora labelled with dialog state
  - DSTC1: bus route information in Pittsburgh
  - DSTC2: changing user goals, tracking requested slots, related to restaurant search
  - DSTC3, DSTC4 and DSTC 5
- Dataset
  - Input consists of list of turns
    - Output (system): transcript, dialog acts
    - Input: asr-hyps, slu-hyps, batch asr (hyps, cnet, lattice)

Previous Work

- Generative models
  - Hidden user goals generate observations (SLU hypothesis)
  - Horvitz and Paek, 199; Williams and Young 2007; Young et al., 2009; Thomson and Young, 2010
- Discriminative models
  - MaxEnt to estimate probability that hypothesis is correct
  - Better performance than generative models
  - Bohus and Rudnicky (2006), Henderson et al (2013), Lee and Eskanazi, 2013
- Web Ranking approach
  - Handwritten rules to generate possible hypothesis
  - Use regression ranker to get the best
  - Williams (2014)

Approach: system architecture

- State graph
    - Nodes are dialog states
    - Arcs are rules that generate state hypothesis given current state and turn data
- Decoder
    - Start from initial state
    - Query graph for list of next state hypothesis
    - Use pair-wise ranker to order hypothesis
    - Prune bottom hypothesis
- Pairwise ranker
    - Classifier that estimates if X>Y
    - Can use features that are relative to specific pair: difference in confirmed slot counts

Oracle Ranker

- Estimates ideal order of states for training
  - Estimate F1 score of state by comparing it to labeled data (goal labels, requested labels and method)
  - Sort states by their F1 score
- Useful for improving State Graph rules
- Accuracy ceiling analysis
  - Oracle accuracy is not 100%
  - SLU and ASR might not have correct hypothesis

Results: Oracle and Baseline

		Joint Goals	Requested	Method
BASELINE	Accuracy	**0.6120959**	**0.893617**	**0.83032**
	l2	0.631869	0.1743412	0.2658
	roc.v2_ca05	0	0.0004036	0.33738
ORACLE	Accuracy	**0.786757**	**0.9870177**	**0.88826**
	l2	0.626446	0.1141163	0.35708
	roc.v2_ca05	0.0003313	0.0003654	0.076

References

- Horvitz, E., & Paek, T. (1999). A computational architecture for conversation. *Courses and Lectures-International*
- Williams, J. D., & Young, S. (2007). Partially observable Markov decision processes for spoken dialog systems. *Computer Speech and Language*, *21*(2), 393–422.
- Young, S., Gasic M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., & Yu, K. (2010). The Hidden Information State model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language*, *24*(2), 150–174.
- Thomson, B., & Young, S. (2010). Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech and Language*, *24*(4), 562–588.
- Bohus, D., Rudnicky, A. I., & Rudnicky, A. (2006). A " K Hypotheses + Other " Belief Updating Model A " K Hypotheses + Other " Belief Updating Model, 1–6.
- Henderson, M., Thomson, B., & Young, S. (2013). Deep Neural Network Approach for the Dialog State Tracking Challenge. *Proceedings of the SIGDIAL 2013 Conference*, 467–471.
- Williams, J. D. (2014). Web-style ranking and SLU combination for dialog state tracking. *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, (June), 282–291.
- Henderson, M., Thomson, B., & Williams, J. (2013). Dialog State Tracking Challenge 2 & 3, (September), 1–22.

hazel

A Recipe Reader

Lauren Fox, Maria Sumner, Elizabeth Cary

Overview

- Challenge
- Tools
- Functionality
- Grammar
- Sample interaction
- Issues and Successes
- Demo

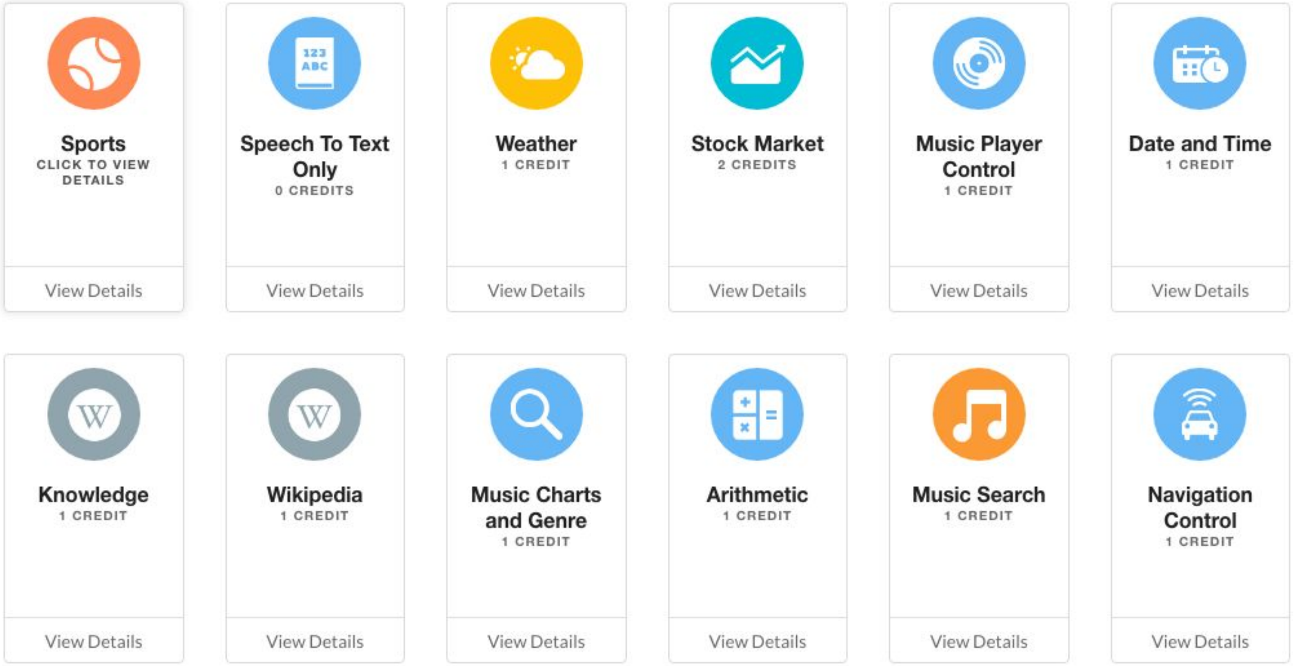
Challenge

- Follow a recipe without:
  - Touching computer/cookbook
  - Referring back to text
- Add additional support as needed
- Improve on existing systems

Tools

- Implemented in Python:
  - Houndify
  - BeautifulSoup - HTML scraper -> Allrecipes.com
  - Google tts
  - Sox

Houndify's public domains



Sports
CLICK TO VIEW DETAILS
View Details

Speech To Text Only
0 CREDITS
View Details

Weather
1 CREDIT
View Details

Stock Market
2 CREDITS
View Details

Music Player Control
1 CREDIT
View Details

Date and Time
1 CREDIT
View Details

Knowledge
1 CREDIT
View Details

Wikipedia
1 CREDIT
View Details

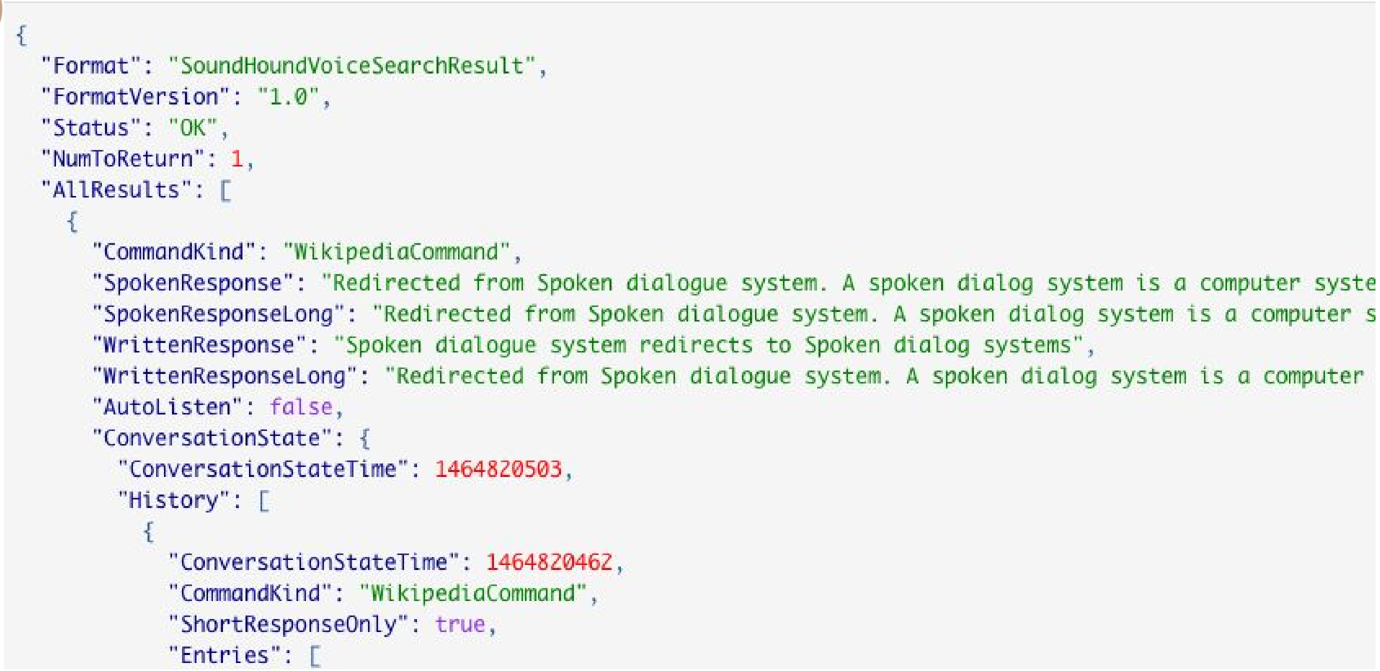
Music Charts and Genre
1 CREDIT
View Details

Arithmetic
1 CREDIT
View Details

Music Search
1 CREDIT
View Details

Navigation Control
1 CREDIT
View Details

Response JSON



```
{
  "Format": "SoundHoundVoiceSearchResult",
  "FormatVersion": "1.0",
  "Status": "OK",
  "NumToReturn": 1,
  "AllResults": [
    {
      "CommandKind": "WikipediaCommand",
      "SpokenResponse": "Redirected from Spoken dialogue system. A spoken dialog system is a computer syste
      "SpokenResponseLong": "Redirected from Spoken dialogue system. A spoken dialog system is a computer s
      "WrittenResponse": "Spoken dialogue system redirects to Spoken dialog systems",
      "WrittenResponseLong": "Redirected from Spoken dialogue system. A spoken dialog system is a computer
      "AutoListen": false,
      "ConversationState": {
        "ConversationStateTime": 1464820503,
        "History": [
          {
            "ConversationStateTime": 1464820462,
            "CommandKind": "WikipediaCommand",
            "ShortResponseOnly": true,
            "Entries": [
```

Functionality

- Read ingredients
  - Double recipe
  - Next
  - Back
  - Repeat
  - Substitutions
- Read directions
  - Next
  - Back
  - Repeat
  - Set timer

- Answer Questions
  - Open domain

"How many calories are there in butter?"

"What's the weather like in Seattle?"

"What is a spoken dialogue system?"

"How many tablespoons in a cup?"

"What's the capital of Ireland?"

Grammar

([[("what\'s"|("what"."was")|("go".["back"]."to"))."the"]."step"]."before")

Possible matches:

“What’s the step before” ; “Go back to the step before” ; “Step before” ; “Before” ; “What was the step before”

Sample Custom Grammar:

```
clientMatches = [ {
      "Expression" : '(("next".["step"]) | ("what\'s"."next") |
      ("what"."do"."i"."do".("next"|("after".[("this"|"that")])))
      |("go"."forward"."a"."step"))',
      "Result" : { "Intent" : "NEXT" },
      "SpokenResponse" : "Next step.",
      "SpokenResponseLong" : "Okay, going to the next step.",
      "WrittenResponse" : "Next step",
      "WrittenResponseLong" : "Going to the next step"
      } ]
```

Sample Interaction

Hazel: 1 cup butter, softened, 1 cup white sugar, 1 cup packed brown sugar

User: Substitution

Hazel: What would you like a substitution for?

User: Brown sugar

Hazel: You can substitute 1 cup packed brown sugar for 1 cup white sugar plus ¼ cup molasses and decrease the liquid in recipe by ¼ cup, or...

Issues and Successes

- Successes
    - Universality - Allrecipes.com
    - Added features
    - Use of Houndify’s domain for conversions, nutrition information
- Issues
    - Add ingredient amounts in directions
    - Lack of barge-in
    - Restricted to domain
    - Skip to specific step

Demo

http://students.washington.edu/carye/demo.html

References and Resources

References:

- Chu-Carroll, J., & Brown, M. K. (1997, July). Tracking initiative in collaborative dialogue interactions. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics* (pp. 262-270). Association for Computational Linguistics.

- Nass, C. & Lee, K. (2001). Does computer-synthesized speech manifest personality? Experimental tests of recognition, similarity-attraction, and consistency-attraction. Journal of Experimental Psychology: Applied, 7(3), 171-181.

Resources with links:

- Houndify
- Google tts
- BeautifulSoup
- Sox

Investigating the Role of the Reparandum in Speech Disfluencies

George Cooper

What are disfluencies?

- Disfluencies include filled pauses, repetitions, and corrections
- Parts:
  - Reparandum (the part that’s corrected or repeated)
  - Editing phase (filler words)
  - Repair (the correction or repetition)

Example disfluency

We had $\underbrace{\text{the cat}}_{\text{reparandum}}\underbrace{\text{, uh,}}_{\text{editing phase}}\ \underbrace{\text{the dog}}_{\text{repair}}$ first

What's in a reparandum?

- NLP systems usually remove the reparandum
- Ferrara et al. (2004) suggests that a reparandum can set expectations about what comes after the repair

What’s the next word?

- Sentence A: “I found it at the corner…,”
- Sentence B: “I found it at the grocery, I mean the corner…”

Finding “grocery store” reparandums

- Disfluencies drawn from the Switchboard section of the Penn Treebank corpus
- Repetitions and restarts removed
- Evaluated probabilities using US English generic language model from CMU Sphinx

A tale of two probabilities

- $p_{\text{reparandum}}$: The probability of the next word when the repair is removed
- p_{repair}: The probability of the next word when the reparandum is removed

Examples of the probabilities

I found it at $\underbrace{\text{the grocery,}}_{\text{reparandum}}$ $\underbrace{\text{I mean}}_{\text{editing phase}}$ $\underbrace{\text{the corner}}_{\text{repair}}$ store.

- $p_{\text{reparandum}} = p(\text{store} \mid \text{I found it at the grocery})$
- $p_{\text{repair}} = p(\text{store} \mid \text{I found it at the corner})$

Results

condition	# disfluencies	% disfluencies
$p_{reparandum} < p_{repair}$	4686	31.0%
$p_{reparandum} = p_{repair}$	8976	59.5%
$p_{reparandum} > p_{repair}$	1424	9.4%
	15087	**100%**

The real “grocery store” reparandums

- I reviewed the 100 disfluencies with the highest preparandum – prepair
- In only ten cases did the reparandum seem more informative than the repair
- Best example:

I don’t think we’ve missed a fish store on the entire $\underbrace{\text{east}}_{\text{reparandum}}$, $\underbrace{\text{northeast}}_{\text{repair}}$ coast of the United States.

Next steps

- Deeper analysis of the disfluencies with the most informative reparandums
- Analysis of incomplete words at the end of reparandums

Spoken Dialog Systems and the Wikipedia API

LAURIE DERMER – NATE PERKINS – KATHERINE TOPPING

Wikipedia API(s)

Wikipedia API(s)

lots of API variants

focused on

1. finding documents given search params
2. identifying sections given document title
3. looking up text for section given document title

Example : https://en.wikipedia.org/w/api.php?action=query&prop=extracts&exinfo&section=1&titles=Albert+Einstein

Web Service

Wanted to keep code for interacting with Wikipedia API out of VXML
- to minimize awkward XML/Javascript development environment

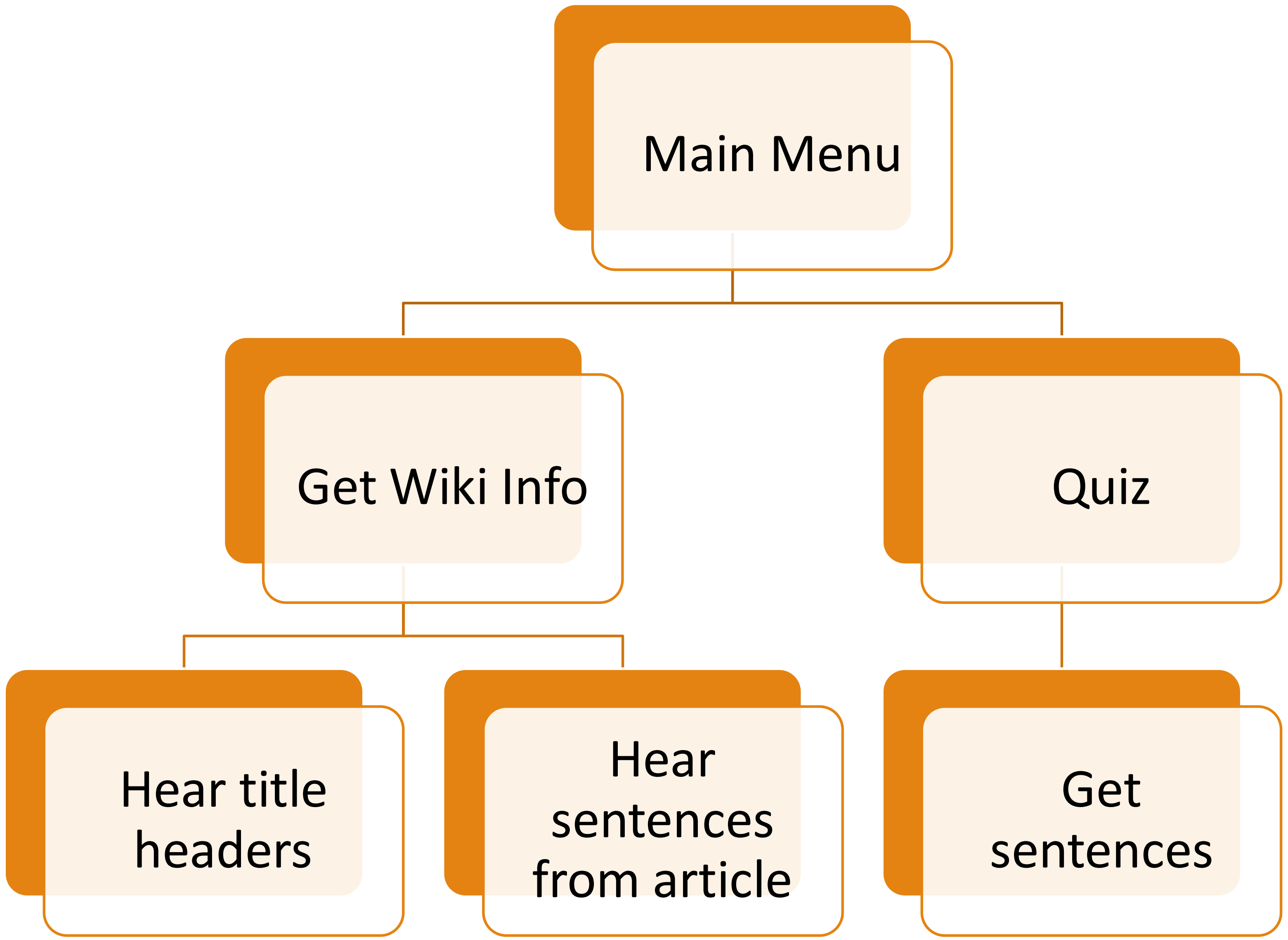
Used Python via CGI

Built API to consume CGI params and produce XML responses

Example :
- Input : https://students.washington.edu/nperk/wiki_test.cgi?action=Search&search=Albert+Einstein
- Output -->

```
<titles>
<title>Albert Einstein</title>
<title>Albert Einstein Medal</title>
<title>List of things</title>
<title>Albert Einstein Award</title>
<title>Albert Einstein Peace Prize</title>
<title>Max Planck Institute</title>
<title>Albert Einstein Memorial</title>
<title>Albert Einstein House</title>
<title>Albert Einstein School</title>
<title>Albert Einstein World Award of
Science</title>
</titles>
```

Call Format and VoiceXML


Main Menu
Get Wiki Info
Quiz
Hear title headers
Hear sentences from article
Get sentences

Call Format

Users can simply ask for information about a historical figure

- System uses Wikipedia API to retrieve a few sentences about the given historical figure

Or, users can ask to "quiz" the system

- User specifies which historical figure they want to quiz the system on
  - Limited to (female) artists and computer scientists, as well as (male) inventors and politicians
  - System will ask user to specify what "type" of person the historical figure was/is
- The system gives a response created by a gibberish generator

VoiceXML

System-initiative, though users can go "off script" with quizzing and some fun easter eggs

- "Why are you a woman?"

Queries limited to historical figures due to having to implement a grammar

- Grammar can't be infinite, even though I'm a very fast typer

Responses limited to a few sentences

- Nobody wants to listen to a robot talk forever

VoiceXML

Using the <data> tag in VoiceXML in order to retrieve data from Wikipedia

- "Search" to return a correct corresponding Wikipedia article title (i.e. "Frank_Underwood_(House_of_Cards)" as a title for "Frank Underwood" as input)
- "TitleSections" to return section names in an article
- "TitleText" to return the initial "intro" text in an article

<data> calls external cgi script (also stored on Vergil) instead of using JavaScript inside VoiceXML

- Much cleaner
- Don't have to deal as much with VoiceXML
- VoiceXML is awful

Ouput is generated by <data> returning the intro text of the selected Wikipedia article ("TitleText")

Extras (for Added Flavor)

System's name is Ada

- This will likely change

She's very polite

- This could also change

She gets flustered when you turn the tables and ask to quiz her

- She admits to having studied only using Wikipedia
    - (Shock! Horror!)
    - System can't pronounce "Wikipedia"

(206) 316-8757

Generating Random Text

Basic Idea

We wanted text that sounded like something that COULD make sense, but is also obviously not true.

There are some toys on the web that do some similar things:

- 'college crapplication' - generates from a corpus of college admission essays
- the 'subreddit simulator', if you've heard of that – it mimics the style of user responses in certain forums and has conversations with itself.
- some Twitter bots too, which I'll talk about later (DeepDrumpf, DeepLearnTheBern)

The first two examples use a Python library called 'markovify' to build a model and generate text.

Markovify

on the web at https://github.com/jsvine/markovify/

Very easy to use:

```
import markovify
text = corpus_file.read()
text_model = markovify.Text(text)
sentence= text_model.make_sentence()
```

PyCharm auto-installed it for me, or you could "pip install markovify"

Corpus

We created our own mini-corpus to seed our model.

We created four categories ("artist", "computer scientist", "inventor", "politician")

- Kept them consistent for gender within each category to keep pronouns consistent in the models
- Each category has one corpus file
- Used internet lists to get about 20-25 names per category
- Got a bunch of first paragraphs from the Wikipedia API

Markovify generates individual sentences out of the box, but we wanted paragraphs.

- So we used regex to get replace periods with PUNCT. Fooled you, Markovify!

Corpus format

Used regex to remove anything between parentheses or brackets (common in Wiki first paragraphs)

We replaced full names with FULLNAME, then first names with FNAME, then last names with LNAME.

If a person has more than two names in their name, the last name regex first tries to match all of the non-first names, but then defaults to the last single word in their name.

```
FULLNAME was a Scottish-born scientist,
inventor, engineer and innovator who is
credited with patenting the first practical
telephone PUNCT

FULLNAME, was an American botanist and
inventor PUNCT The exact day and year of his
birth are unknown; he was born into slavery
in Missouri, either in 1861, or January 1864
PUNCT

FNAME Finley Breese LNAME was an American
painter and inventor PUNCT After having
established his reputation as a portrait
painter, in his middle age LNAME contributed
to the invention of a single-wire telegraph
system based on European telegraphs PUNCT He
was a co-developer of the LNAME code, and
helped to develop the commercial use of
telegraphy PUNCT
```

Generated sentences

The paragraph generator takes any full name ("Ada Lovelace") and a category ("artist" etc, or "f", "m", or "all") and generates sentences as if they're about that person.

PUNCT gets changed back to "." when the text is generated.

Before the system generated full paragraphs, it would sometimes return "first sentence"-looking sentences in the middle of a paragraph

Because it generates a full paragraph at a time, last name/pronoun references are based on the model and feel more natural.

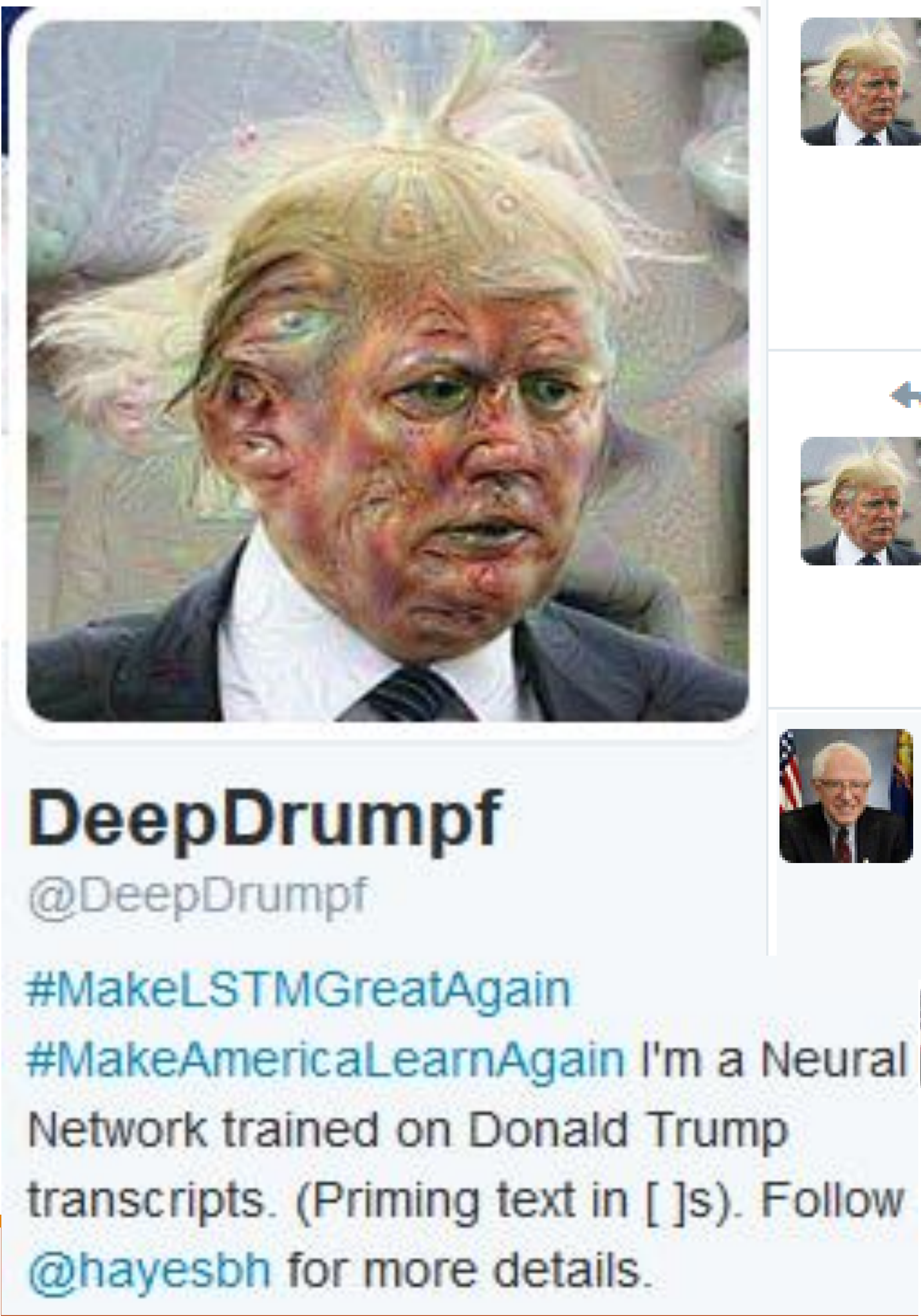
More corpus makes better quality generated text!

```
Ada J. Lovelace is an American computer
scientist, with research interests in
theoretical computer science and formal
methods. She is the Chief Technology Officer
at One Medical Group. Previously, she was the
first successful high level programming
language.

Augusta Ada King-Noel, Countess of Lovelace
was a French-American artist. Best known for
her theoretical prediction concerning the
existence of the University of California,
San Diego.

Ada Lovelace is a Japanese multimedia artist,
singer, songwriter, and peace activist who is
a Cuban-American abstract, minimalist
painter. She turned 100 in May 2015.
```

Similar Bots

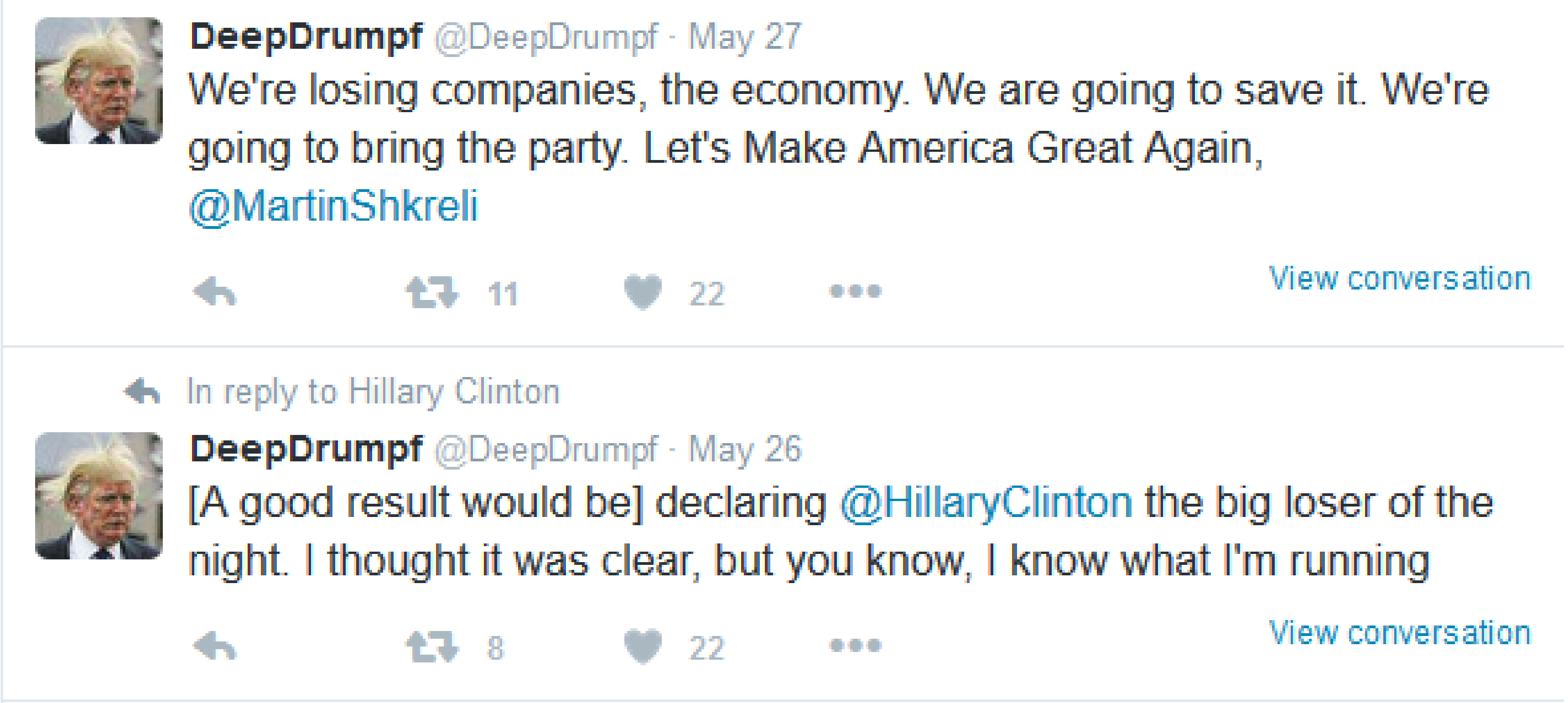
BUT THAT USE A NEURAL NETWORK

AND TWITTER



DeepDrumpf
@DeepDrumpf

#MakeLSTMGreatAgain #MakeAmericaLearnAgain I'm a Neural Network trained on Donald Trump transcripts. (Priming text in [ ]s). Follow @hayesbh for more details.

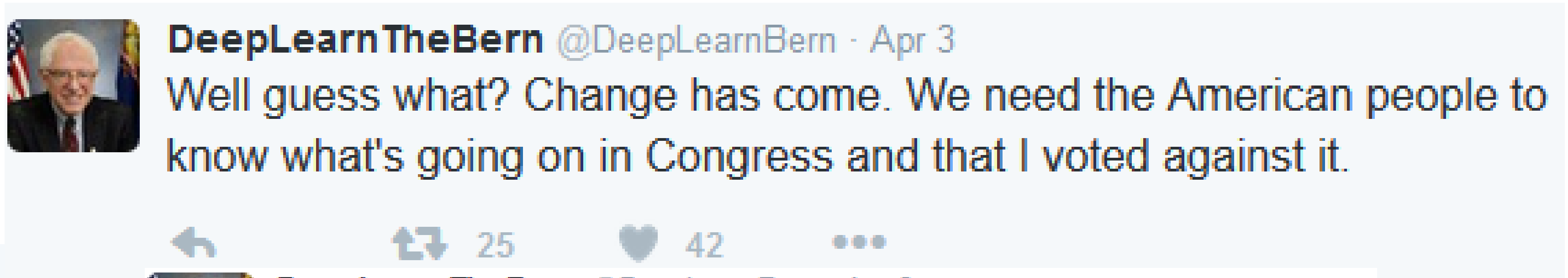


DeepDrumpf @DeepDrumpf · May 27
We're losing companies, the economy. We are going to save it. We're going to bring the party. Let's Make America Great Again, @MartinShkreli

11 22
View conversation

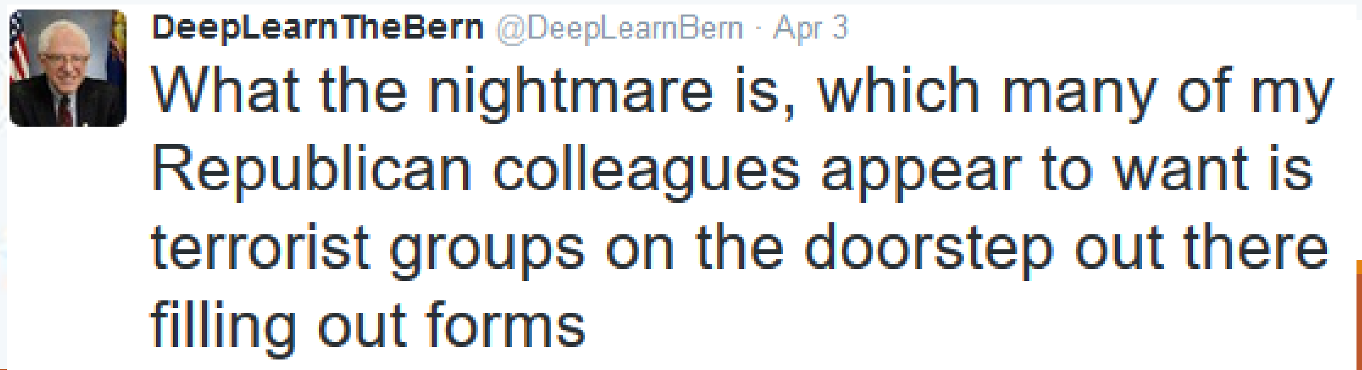
In reply to Hillary Clinton
DeepDrumpf @DeepDrumpf · May 26
[A good result would be] declaring @HillaryClinton the big loser of the night. I thought it was clear, but you know, I know what I'm running

8 22
View conversation



DeepLearnTheBern @DeepLearnBern · Apr 3
Well guess what? Change has come. We need the American people to know what's going on in Congress and that I voted against it.

25 42



DeepLearnTheBern @DeepLearnBern · Apr 3
What the nightmare is, which many of my Republican colleagues appear to want is terrorist groups on the doorstep out there filling out forms

The Adventure of the Prosodic Corpus

John T. McCranie
jtm37@uw.edu
LING575

02/Jun/2016

overview

- 2nd lang acquisition, CALL, curriculum development
- corpus linguistics
- learner corpora
- prosody in learner corpora?
- the LeaP corpus

example corpora

monolingual:

- American National Corpus - 22M words, written and spoken, letters, essays, court transcripts
- British NC - 100M words
- Türkçe Ulusal Derlemi - 50 milyon sözcükten oluşan

multilingual:

- OPUS project - open source online parallel snippet translations

learner:

- MERLIN - language level reference (CERF: A1-C3)
- ICLE - ongoing, 16 native languages, 3.7M words, standardized contribution criteria, parallel related corpora

learner corpora

metadata:

- subject background: native language, proficiency level in target language, other languages known, curriculum
- production types: essay, reading, dialog

markup:

- POS tagging, syntax trees
- transcription of recording
- error identification

the LeaP corpus

from the LeaP manual:

- part of *Learning Prosody in a Foreign Language Project*
- recordings of learners of two targets: German and English
- reading and retelling stories, reading word lists, interviews
- 131 speakers of 32 native languages, with about 12 hours of recordings
- comes with a set of analysis tools
- wide variations in speakers backgrounds: age, sex, native languages, level of competence, length of exposure to the target language, age at first exposure to the target language, motivation, musicality

research questions

1. can we extract feature sets from the LeaP corpus to correlate prosodic element usages with learner levels?
2. can we compare phrase level pitch accent usage across proficiency levels?

look before you LeaP

each work in the collection consists of 3 files: wav, textgrid, xml

however:

- tools no longer available
- overgeneralized file format
- missing data

language counts

Language	Native	Other
Abore	0	1
Anyi	2	0
Arabic	3	0
Bosnian	2	0
Bulgarian	1	0
Chinese	8	2
Czech	1	0
Dutch	0	2
Edo	1	0
Efik	1	1
English	8	44
French	3	40
German	26	17
Hungarian	2	1
Ibibio	2	0
Igbo	1	1
Italian	6	4
Japanese	0	2

Language	Native	Other
Kazakh	0	1
Korean	5	1
Koyo	1	0
Kurdish	1	0
Maori	0	1
Persian	1	0
Polish	8	1
Portuguese	0	2
Romanian	3	0
Russian	9	10
Serbian	0	1
Slovak	1	1
Spanish	5	13
Swedish	0	1
Thai	1	0
Turkish	2	2
Ukrainian	1	1
Welsh	0	1
Yoruba	1	0

phrase level pitch accent

User	Target	Level	Tone	Phrase	Count
ai	English	native	^H*	IN NN	1
ax	English	unknown	H%	IN NN	1
be	English	unknown	L*	IN NN	1
br	English	unknown	^H*	IN NN	1
bu	English	unknown	H*	IN NN	1
bu	English	unknown	L*	IN NN	1
bu	English	unknown	L*+H	IN DT NN	1
bv	English	unknown	H*	IN NN	1
bz	English	unknown	H*	IN NNS	1
ca	English	native	L+H*	IN DT NN	1
cb	English	native	L+H*	IN NN	1
cb	English	native	L+!H*	IN NN	1
cd	English	other	H*	IN NN	1
cl	English	unknown	L*+H	IN NN	1
cl	English	unknown	L*+H	IN DT NN	1

conclusions

add prosody to an existing corpus, such as ICLE. it is growing, uses consistent data collection methods.

UNIVERSITY *of* WASHINGTON

Fill-in-the-Blank Coreference Generation

Ethan Roday

LING 575 SP 16

BACKGROUND

WHAT IS THE PROBLEM?

Given a mention in context, predict its form

> Predict over human-human data to build human-like systems

Constraints:

- Generative
  - Predicts the *form* of a reference, not the referent
- Incremental
  - No access to future data

APPROACH

Two tasks:

- Pronoun vs. Non-Pronoun
- Pronoun vs. Reduced Reference vs. Full Reference

Framing as an ML problem with three feature sets:

- Reference chain features
  - Previous mentions, length of chains, etc.
- Local features
  - Sentence position, grammatical role, etc.
- Global features
  - Discourse structure, number of other referents, etc.

DATASET

- CallHome subset of OntoNotes corpus
  - 142 conversations
  - 2,600 coreference chains
    - 1,414 of length > 2
  - 16,879 mentions (i.e. instances)
- Open domain
- Richly annotated:
  - Coreference
  - NEs
  - Penn Treebank-style parses
  - PropBank annotations

PRELIMINARY RESULTS

> Two-way classification
> Reference chain features only

Model	Accuracy
Always Pronoun	0.558370
New/Old Only	0.762452
All Chain	0.782248
Khudyakova Best*	0.899
Khudyakova Worst*	0.796

> *HUUUGE CAVEAT: Not the same dataset

Analyzing Frustration in SDS Grounding Methodologies

Micaela Tolliver

Analyzing Frustration in SDS Systems

- Prosody-based Automatic Detection of Annoyance and Frustration in Human-Computer Dialog - Walker et. al, 2002
    - Frustration annotations utilized for automatic recognition
- Communicator 2000 corpus
    - Annotations for:
        - Emotions
            - Neutral, Annoyed, Amused, Angry, Disappointed, None, Not Applicable
            - 14.69% of user responses were negative emotions (Annoyed, Angry, Disappointed)
        - Dialog Acts
        - Other information

Previous Analysis on Frustration

- “The impact of response wording in error correction subdialogs” - Goldberg and Kirchoff
  - Analyzed the Communicator 2000 Corpus
  - Error Correction dialogs
    - Repeating previous phrase
    - Rephrasing the prompt
    - Apologizing
  - Rephrasing the prompt and apologizing provides less frustration and less word error rates

Grounding

- Grounding based off of Confirmation Strategies and Speech Acts
  - Confirmation strategies repeat user input to avoid errors; recognizing this as a grounding strategy
- Explicit Confirmation
  - You are departing from Seattle. Is this correct?
- Implicit Confirmation
  - Where would you like to go from Seattle?
- Acknowledgement
  - Great! I am adding the flights to your itinerary.

Grounding (Cont.)

- Tagged recognized information:
  - "I want to fly to Seattle" -> I want to fly to <CITY> Seattle </CITY>
  - When would you like to fly to Seattle? -> When would you like to fly to <CITY> Seattle </CITY>?
- Used this tag information to recognize different grounding methods: compared the previous user utterances' tags that are recognized by ASR to next system utterance tags
  - System does not repeat the previous tags
  - System repeats all of the previous tags
  - System repeats some of the previous tags
  - System repeats more than just the previous tags (Information from earlier in the dialog)

Current Results - Emotion Rates

Grounding Type	annoyed	annoyed/frustrated	disappointed/tired	Overall Bad
Implicit Confirmation	0.0909	0.0158	0.0074	0.1141
Explicit Confirmation	0.1604	0.0232	0.0089	0.1925
Acknowledgement	0.1342	0.0022	0.0067	0.1450
No repeating tags	0.1400	0.0106	0.0042	0.1548
All repeating tags	0.1110	0.0167	0.0042	0.1319
Some repeatings tags	0.0847	0.0169	0.0000	0.1017
More than just previous tags	0.1235	0.0143	0.0143	0.1520

Planned Improvements

- Currently only looking at each methodology (Dialog Act or Tags) individually
- Compare other information in an utterance
  - Example: Do systems back off to explicit confirmation when errors occur?
  - This could really impact how explicit confirmation is perceived
- Look for more/better ways to recognize grounding using recognized tags

Discourse Act Labels as Predictors for Backchannels

A simple model for backchannels in Spoken Dialogs

LING 575: Spoken Dialog Systems
June 2th, 2016

Backchannels in Spoken Dialog

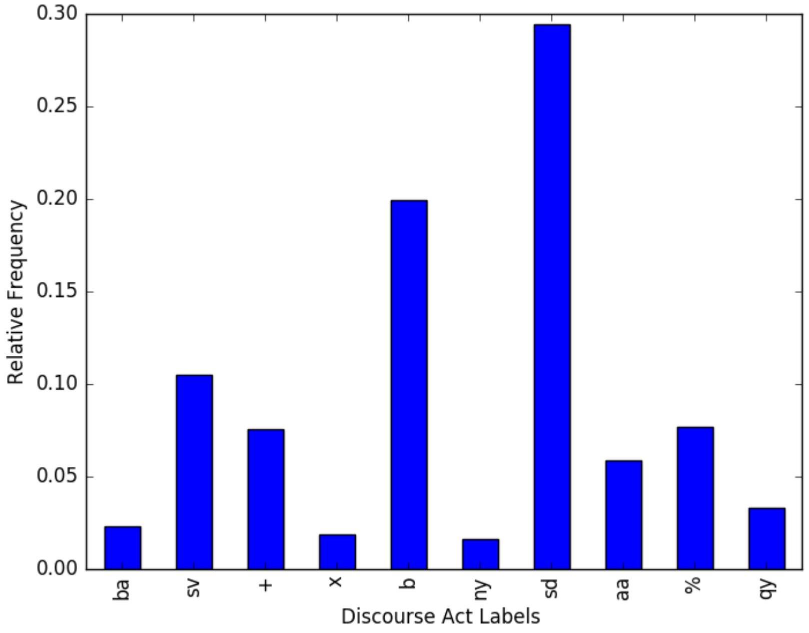
Backchannels (e.g. mm-hmm, Uhuu…) are common elements in a spoken dialog that have important grounding functions, to signal the speaker to continue with her turn.
-> Signal pragmatic completion of a utterance (TRPs)

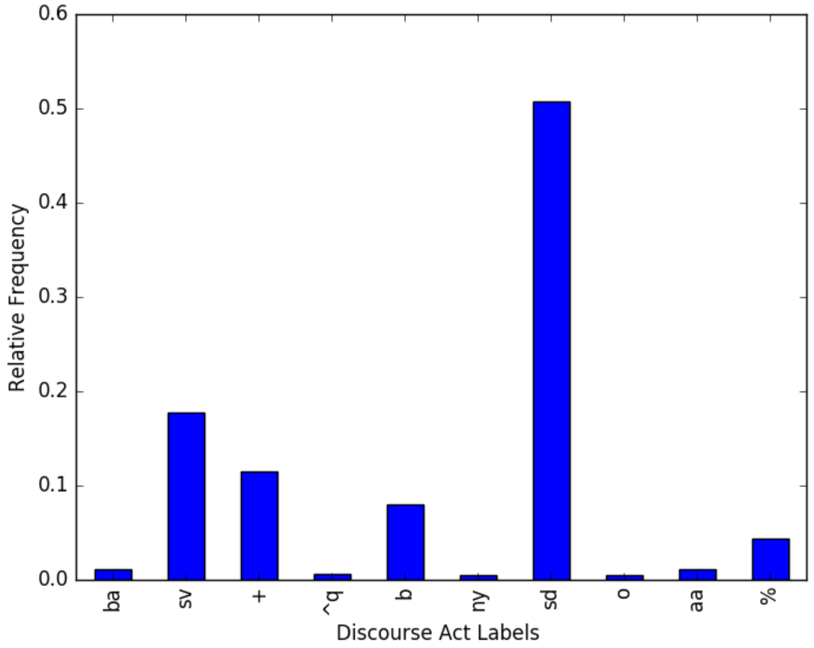
Need to model them into any ASR/G system:

part-of-speech n-grams
pitch, F0 contour o preceding statement
length of preceding utterance

Backchannels as Dialog Acts (DA)

Backchannels can be represented as dialog acts which play an essential role in the discourse structure of a dialog.





Question: Does the consideration of a preceding DA increase the effectiveness of a Backchannel model?

Model and Data

MaxEnt Classification of backchannel and non-backchannel DA when there is a change of speaker

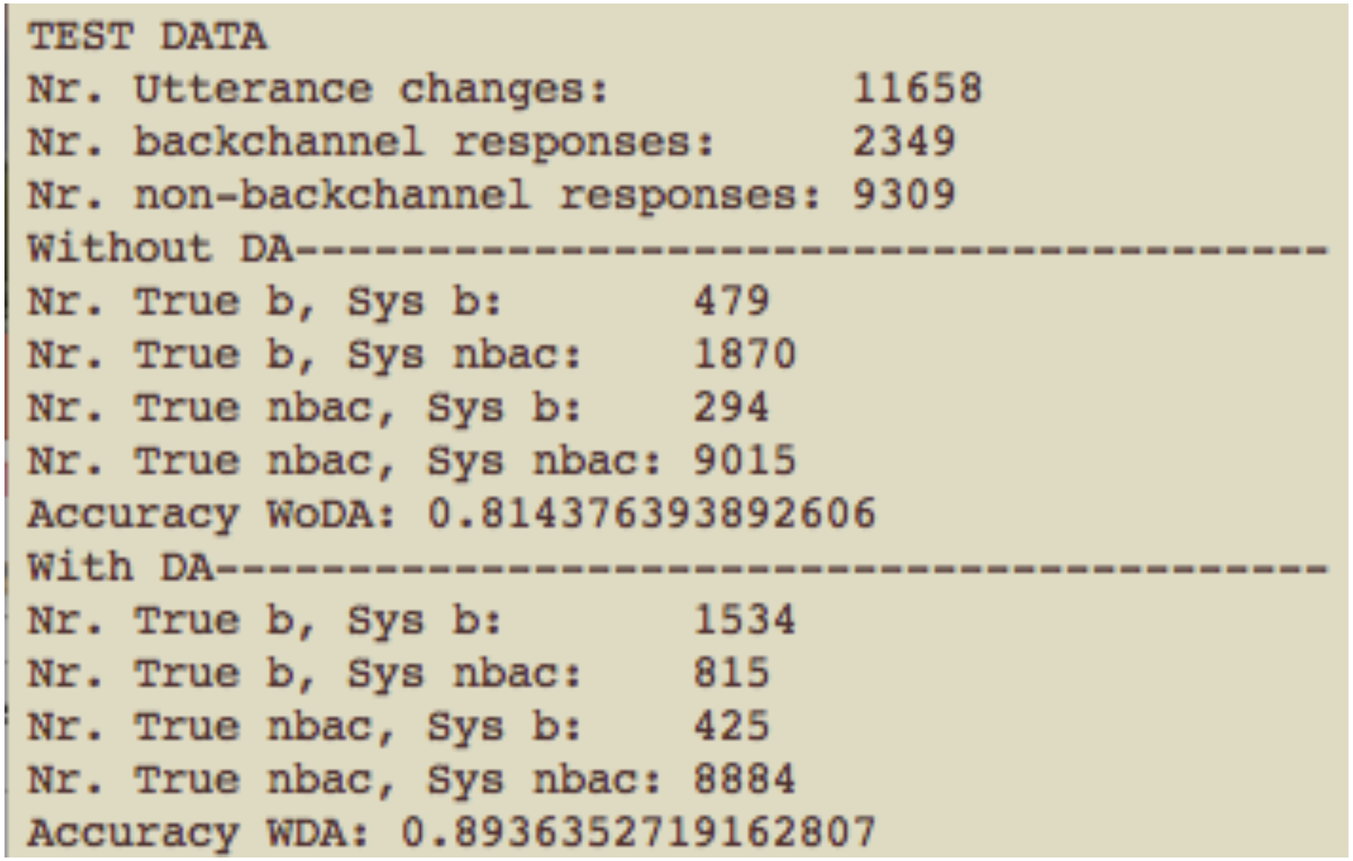
Data: Annotated SWBD-DAMSL

Baseline System: At each change of speaker:
- number of utterances of preceding speaker's turn
- number of words of last preceding utterance

Improved System: Include DA label of preceding last utterance.

Trained model over 104,915 change of turn utterances of SWBD-DAMSL annotated data.

Results



```
TEST DATA
Nr. Utterance changes:          11658
Nr. backchannel responses:      2349
Nr. non-backchannel responses: 9309
Without DA-----------------------------------------
Nr. True b, Sys b:        479
Nr. True b, Sys nbac:     1870
Nr. True nbac, Sys b:     294
Nr. True nbac, Sys nbac: 9015
Accuracy WoDA: 0.814376393892606
With DA--------------------------------------------
Nr. True b, Sys b:        1534
Nr. True b, Sys nbac:     815
Nr. True nbac, Sys b:     425
Nr. True nbac, Sys nbac: 8884
Accuracy WDA: 0.8936352719162807
```

Narrative and Dialog

June 2, 2016

Background

- A lot of recent work by Ruhlemann
  - turn-taking
    - Phenomenon where primary narrator tries to control specific turns, specifically, every third turn
  - co-construction of conversational narrative
    - features of dialog shift depending on which participant is constructing the story
- Using narrative turn taking using the Pear story corpus.

Pear Corpus

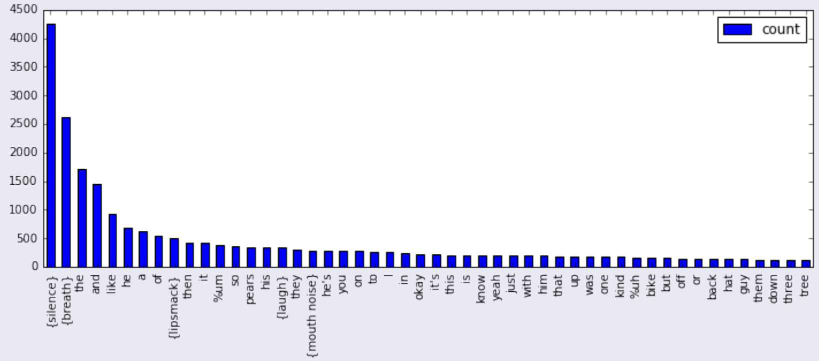
- Initial pair story is a six minute silent film developed by William Chafe at UC Berkeley in 1975. Attempting to keep the study as pure as possible initial participants had similar backgrounds.
- He asked the participants to retell and described the film
- Chafe in his 1981 work indicated that his motive was to explore how different cultures recounted narratives in their language, based on a language-less stimulus
- He later published his work in 1981. (The Pear Stories: Cognitive, Cultural, and Linguistic Aspects of Narrative Production
- Pear story narratives have been published in numerous languages and cultures and extensively analyzed
- "Pear Stories establish that narratives are units of discourse useful for exploring . . . the relationship of cognition, culture, and language–problems which should also be examined in narratives of personal experience told in everyday settings." Shiffirin

Corpus Description

- 20 Files
- Praat format
- Includes Words, phrases, part-of-speech
- All words and phrases contain start and end time
- Speaker and Listener
- special annotations
  - "%": used for uh
  - "{...}": {silence},{breath},{laughter}
  - "^": "^Oral ^Roberts ^University"
  - others include *, -, +, ()

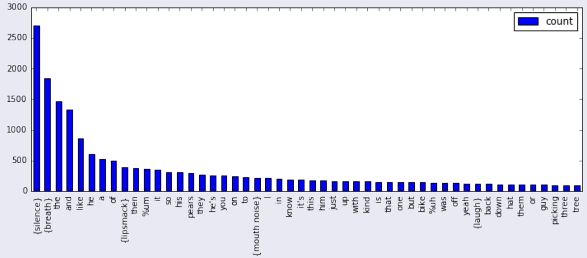
Corpus Distribution

Speaker and Listener

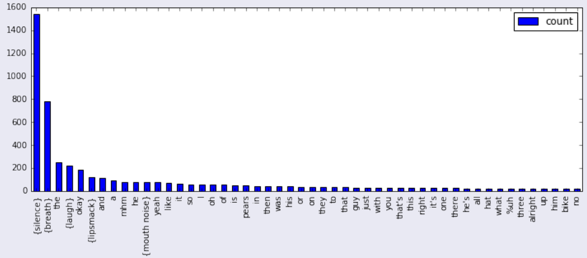


Corpus Distribution

Speaker



Listener



Ordered Turns

1. S (283.989 - 284.575) {silence} {sniff} {silence}
2. L (283.732 - 285.275) {silence} okay {silence}
3. S (282.979 - 284.074) {silence} No, you don't, though, {silence}
4. L (280.913 - 283.905) {silence} I do . No, I do . {silence}
5. S (281.291 - 283.045) {silence} Now, you like want to watch {laugh} this . {silence}
6. L (280.557 - 283.732) {breath} {silence} I do . No, I do .
7. S (279.205 - 281.291) It was weird . {silence} Yeah .
8. L (278.372 - 280.557) Yeah, yeah, okay, I see what you're saying . {silence}
9. S (278.212 - 279.823) colors in it . {silence} It was weird .
10. L (271.538 - 280.391) {silence} Yeah, yeah, okay, I see what you're saying .

Examples 1

- S {silence} Now, you like want to watch {laugh} this . {silence}
- L {breath} {silence} I do . No, I do .
- S {silence} And so then like his hat like blows off of his head . {breath}
- L {laugh} {silence} {breath}
- S bl- – He's wearing a – like a cowboy hat . This is all kind of like cowboy scene . {silence} {laugh}
- L {breath} {silence} {laugh}
- S the guy with the goat leaves and then the guy {silence} goes back up the ladder
- L {laugh} {breath} {silence}
- S {silence} the guy with the goat leaves and then the guy {silence}
- L {silence} {laugh} {breath}
- S {silence} sniff the pears or someth- – I don't know . He like keeps pulling the goat . {silence}
- L {laugh} {silence} {breath}
- S {silence} And %um so then he goes down . And then you know like he's very like taking a lot of time with these pears . Like, {silence}
- L {silence} {laugh} {silence}
- S (erh erh) . Like, it's this loud creak . {breath} {silence}
- L yeah {silence} {laugh}
- S {silence} ˆM- ˆMexican looking guy picking pears . And he has like this handkerchief on . And he keeps like putting them in his like little apron . {breath}
- L {laugh} {breath} {silence}
- S {breath} So at first I thought it was like a war kind of film, {silence}
- L {silence} {laugh} {silence}

Examples 4

- S bomb going off, or something . {breath} So at first I thought it was like a war kind of film,
- L {silence} {laugh} {silence}
- S since you're not accustomed to the noises being over-exaggerated, every time he rips a pear off the tree it sounds like it's like a bomb going off, or something . {breath}
- L {silence} {laugh} {silence}
- S {breath} since you're not accustomed to the noises being over-exaggerated, every time he rips a pear off the tree it sounds like it's like a bomb going off, or something .
- L {laugh} {silence} {laugh}
- S And so it's this guy picking pears, but since you're not like accustomed – {breath} since you're not accustomed to the noises being over-exaggerated, every time he rips a pear off the tree it sounds like it's like a
- L {silence} {laugh} {silence}
- S {breath} And so it's this guy picking pears, but since you're not like accustomed – {breath}
- L {laugh} {silence} {laugh}
- S because it was like I – %eh {silence} I couldn't tell what was going on .
- L {silence} {laugh} {silence}
- S the volume, like the way the film is captured {breath} {laugh}
- L {laugh} {silence} {breath}
- S {silence} %uh these noises . And all the noises of people interacting are way overly exaggerated like {silence}
- L {silence} {laugh} {silence}
- S It's just like {silence} %uh these noises . And all the noises of people interacting are way overly exaggerated like
- L {breath} {silence} {laugh}

References

- Iulian Vlad Serban and Ryan Lowe and Laurent Charlin and Joelle Pineau. *A Survey of Available Corpora for Building Data-Driven Dialogue Systems*. CoRR, 2015.
- Christoph Ruhlemann and Matthew Brook O'Donnell. *Introducing a corpus of conversational stories. Construction and annotation of the Narrative Corpus*. University of Munich, 2012.
- Christoph Ruhlemann. *Narrative in English Conversation: A Corpus Analysis of Storytelling*. Cambridge University Press, 2014.