Clarification in Spoken Dialogue Systems: Modeling User Behaviors

Julia Hirschberg

Columbia University

Acknowledgments

- Svetlana Stoyanchev, AT&T Labs Research
- Sunil Khanal, Alex Liu, Ananta Pandey, Eli Pincus, Rose Sloan, Mei-Vern Then, Jingbo Yang: Columbia University
- Philipp Salletmayer: Graz University of Technology

Speech Recognition in Spoken Dialogue Systems

- Speech Recognition errors in SDS are quite common
  - ~9% in TRANSTAC Speech 2 Speech translation system (for English)
  - ~50% in a deployed system: CMU's Let's Go bus information system

How do SDS Handle Errors?

- Use ASR confidence scores (combination of Acoustic Model likelihood and Language Model posterior probability) to score a recognition hypothesis
- When they believe they have misrecognized a user they use very simple strategies to recover from error
  - Call Andrew Laine.
    - I don’t understand [call Andrew Laine] but would you like me to search the web for it. (Siri)
    - I missed that, could you please repeat?
    - Sorry, could you please rephrase?

How do Humans Handle Errors?

- People typically ask for clarification in very different ways (Williams & Young ‘04; Koulouri & Lauria ‘09)
  - Call Andrew Laine.
    - You want to call whom?
    - Whom do you want to call?
    - Which Andrew do you want to call?
- Termed by Purver ‘04 ***Reprise Clarification Questions:*** Targeted questions that make use of portions of an utterance the hearer believes she ***has*** understood to ask about what she has ***not***
  - 88% of human clarification questions are of this type

Outline

- Building a Dialogue Manager for Speech 2 Speech Translation
- Data Collection for Clarification Questions
- Classification experiments
  - Predicting user behavior
  - Identifying local errors
  - Predicting error type
- Future research

Our Research

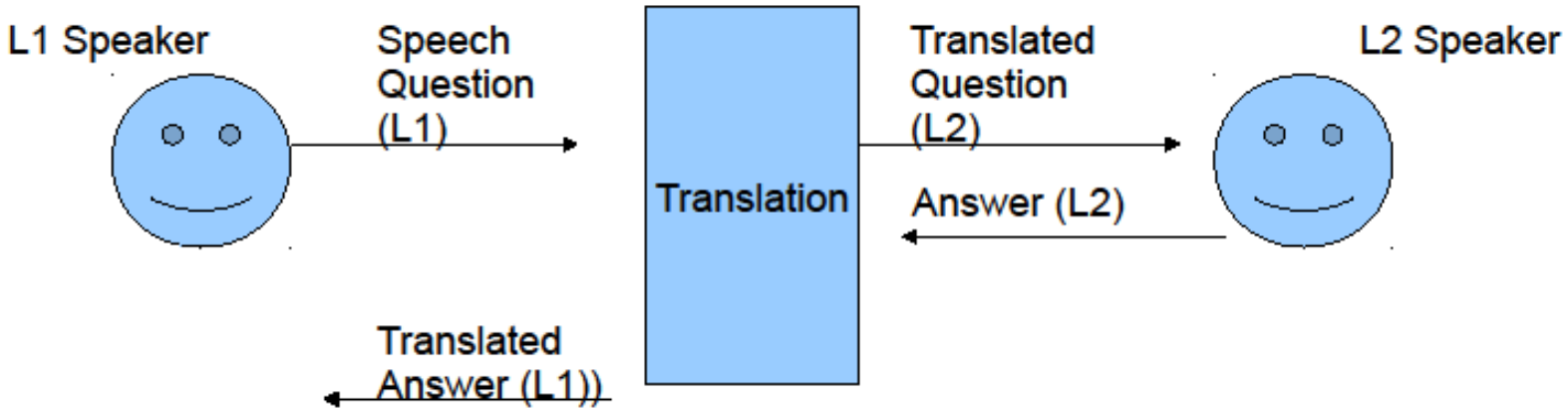
- Study human-human strategies for dealing with Automatic Speech Recognition (ASR) errors in a speech-to-speech translation system (ThunderBOLT)
  - Identify errors that do ***not*** require clarification – where we can guess the meaning or it is not critical
  - Identify clarification strategies for those that ***do***
- Develop methods to detect ***local*** ASR errors with high accuracy
- Create a Dialogue Manager (DM) which can ask appropriate clarification questions when necessary – including Reprise Questions – in interacting with ThunderBOLT users

Clarification in Speech 2 Speech Translation Systems

- DM must support unrestricted conversation between conversational partners who do ***not*** speak one another's language
- ThunderBOLT
  - Supports Speech-to-Speech (S2S) Machine Translation (MT) between American English and Iraqi Arabic
  - DM must identify potential errors in ASR input and try to clarify/correct these before passing transcript to MT

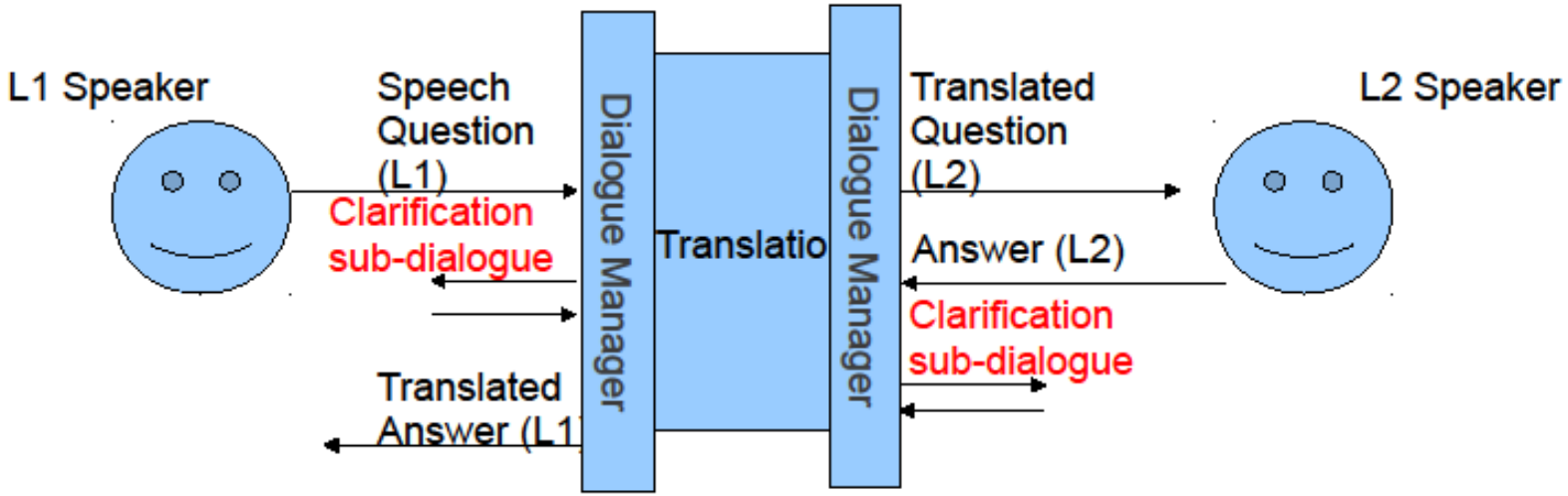
System/Data

- Part of The Broad Operational Language Technology Program (BOLT) DARPA project
- Speech-to-Speech translation system



System/Data

- Part of The Broad Operational Language Technology Program (BOLT) DARPA project
- Speech-to-Speech translation system
- Introduce a clarification component



Corpus

- Speech, ASR and gold standard transcripts from SRI's Iraq-Comm S2S system (Akbacak et al '09)
  - Collected during 7mo of evaluations performed from 2005-08
  - Sample Dialogue (manual transcript/translation)

    English: good morning

    Arabic: good morning

    English: may i speak to the head of the household

    Arabic: i'm the owner of the family and i can speak with you

    English: may i speak to you about problems with your utilities

    Arabic: yes i have problems with the utilities
- Use to collect human clarification questions

Outline

- Building a Dialogue Manager for Speech 2 Speech Translation
- Data Collection for Clarification Questions
- Classification experiments
  - Predicting user behavior
  - Identifying local errors
  - Predicting error type
- Future research

Collecting Clarification Questions

- Approach: collect a text corpus of human responses to ASR transcriptions with missing information using Amazon Mechanical Turk (AMT) crowd-sourcing
- Data: 944 utterances from TRANSTAC corpus which each contain a single ASR error
  - 668 sentences with single-word error segment
  - 276 sentences with multi-word error segment

- Replace errors in transcripts with ‘XXX’
  - Do you own a ?gun? ? → Do you own a XXX?
- Ask 3 Turkers to answer a series of questions about each errorful transcript

Annotator Instructions

How many XXX doors does this garage have

1. Is the ***meaning*** of the sentence clear to you despite the missing word?
2. What do you ***think*** the missing word could be? If you're not sure, you may leave this space blank.
3. What ***type*** of information do you think was missing?
4. If you heard this sentence in a conversation, would you ***continue*** with the conversation or ***stop*** the other person to ask what the missing word is?
5. If you answered "stop to ask what the missing word is", ***what question*** would you ask?

Sample Question1

- Do you own a XXX?

Sample Question1

- Do you own a Hardhat?

Sample Response 1

- Do you own a XXX?
- Turker guesses (word/POS)
  - T1: ? / noun
  - T2: house / noun
  - T3: ? / noun
- Turker proposed clarification questions
  - T1: Do I own a what?
  - T2: ?
  - T3: Do I have what?

Sample Question 2

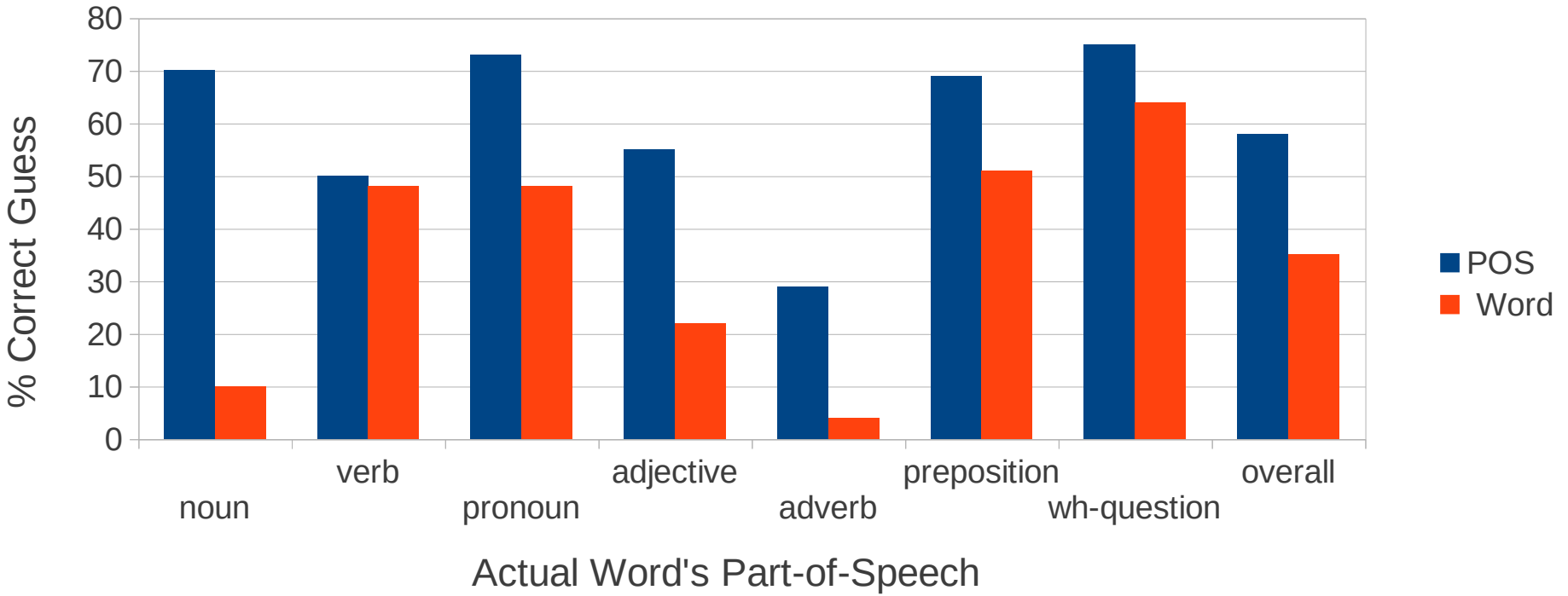
- How long have the villagers XXX on the farm for?

Sample Question 2

- How long have the villagers worked on the farm for?

Sample Response 2

- How long have the villagers XXX on the farm for?
- Turker guesses
  - T1: worked / verb
  - T2: are / pronoun (!)
  - T3: lived / verb
- Turker questions:
  - T 1-3 thought no question was needed



- Users guess correct word 35% of overall cases
- Users guess correct POS tag in 58% of overall cases
- Users are likely to guess a **noun POS** correctly but unlikely to guess the actual word

Possible User Strategies

- For sample input Make sure you close the XXX behind the vehicle
  - Continue without asking a question (infer XXX or inference unnecessary)
  - Stop and ask a question
    - Generic question: What did you say?
    - Confirmation question: Did you mean close the door?
    - Reprise clarification question: What needs to be closed behind the vehicle?

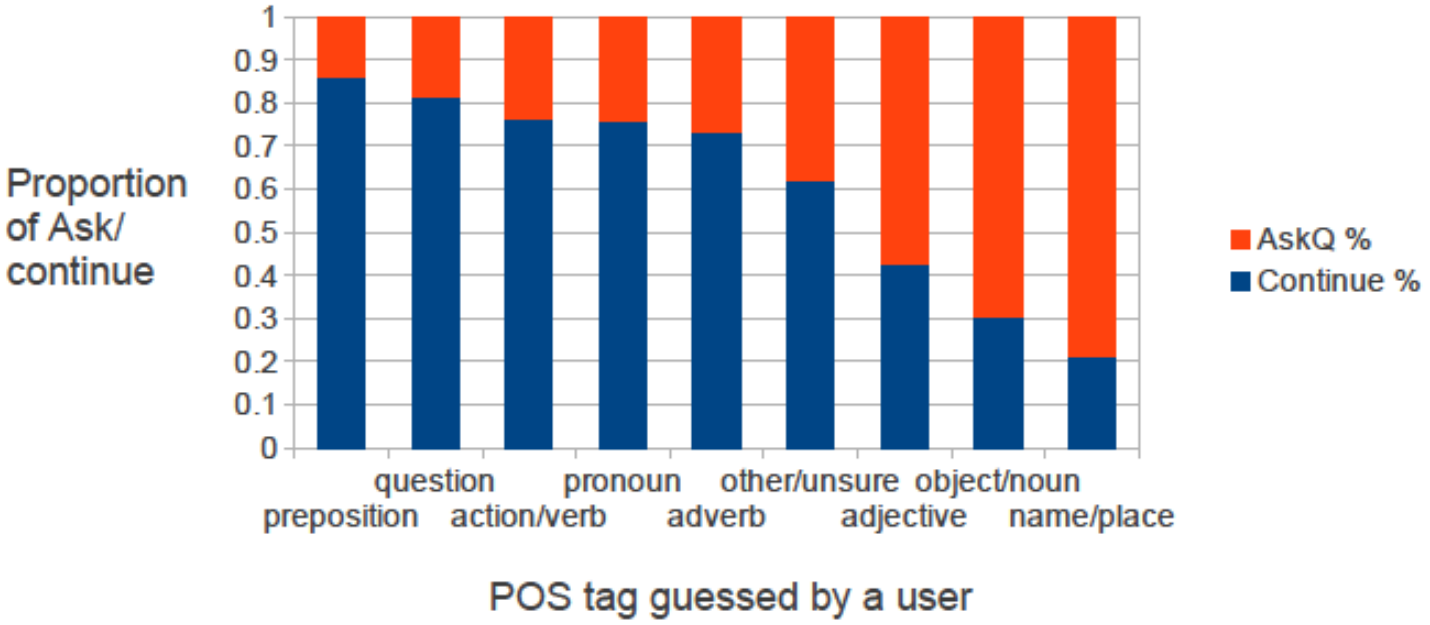
Possible User Strategies

- For sample input Make sure you close the XXX behind the vehicle
  - Continue without asking a question (infer XXX or inference unnecessary) ***62%***
  - Stop and ask a question ***38%***
    - Generic question: What did you say?
    - Confirmation question: Did you mean close the door?
    - Reprise clarification question: What needs to be closed behind the vehicle?

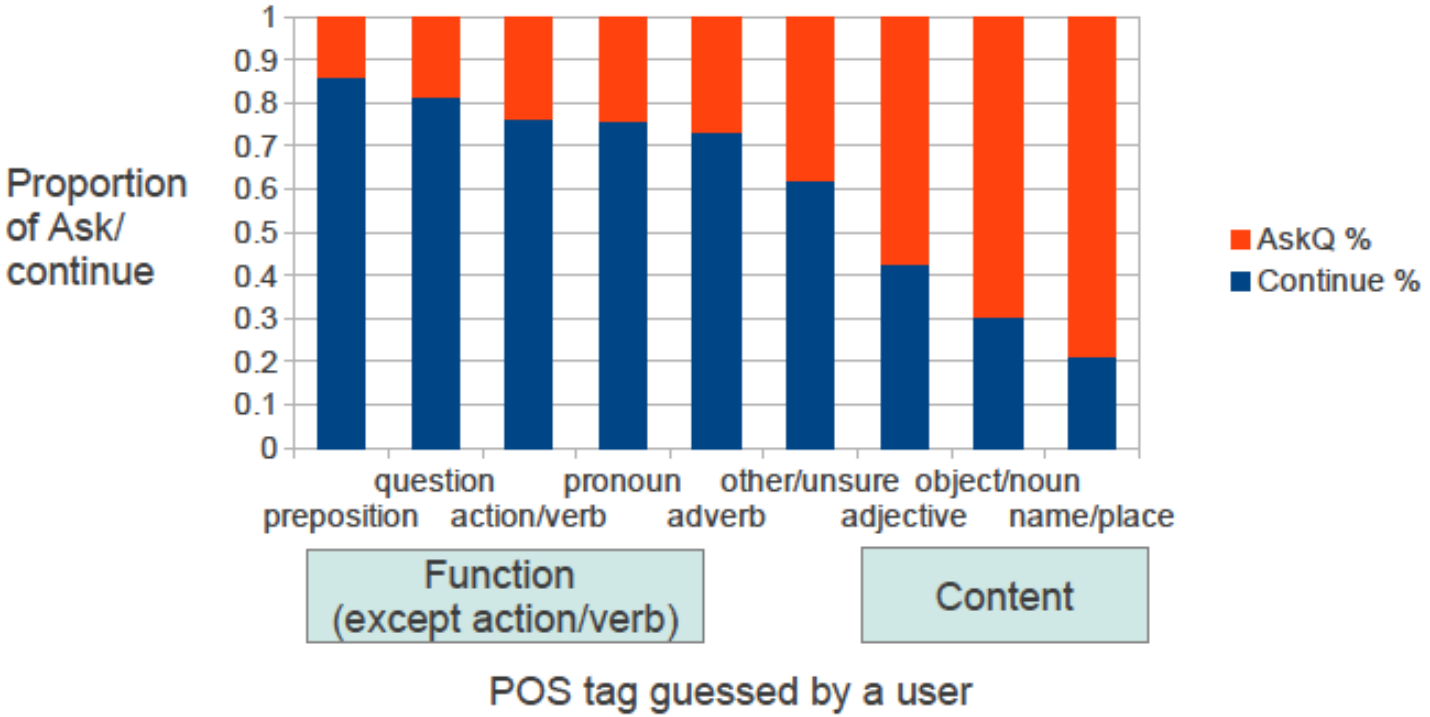
Sample Turker Clarification Questions

do you have anything other than these XXX plans	What plans?
XXX these supplies stolen	What about the supplies?
what else can XXX do if the vehicle don't stop	Can who do?
do you desire to XXX services to this new clinic	To do what about services?
XXX your neighbor reported the theft	Which neighbor?

When do users prefer to NOT ask a question (continue)?



When do users prefer to NOT ask a question (continue)?



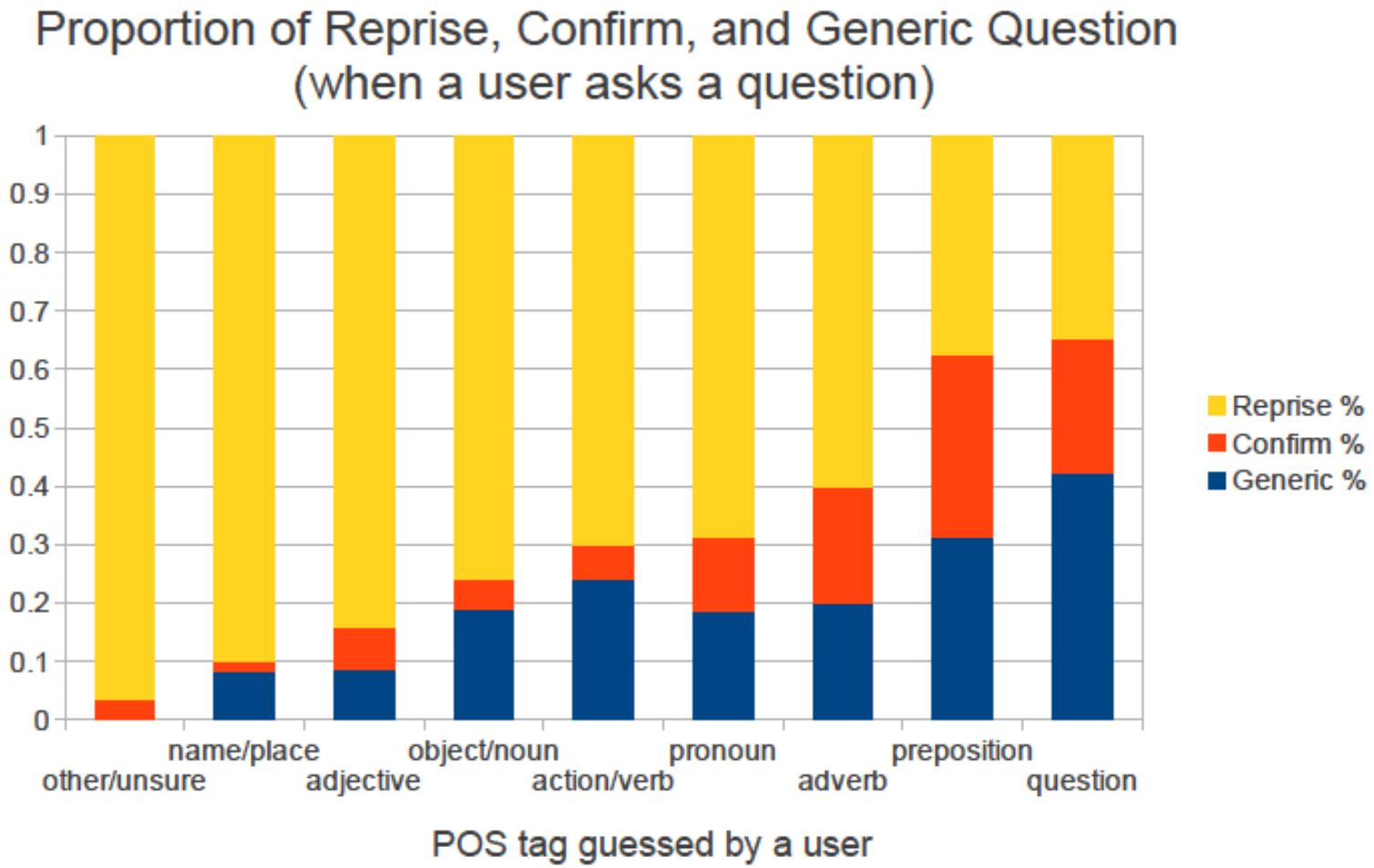
What Types of Questions are Most Frequent?

- For sample input Make sure you close the XXX behind the vehicle
  - Continue without asking a question (infer XXX or inference unnecessary) ***62%***
  - Stop and ask a question ***38%***
    - Generic question: What did you say?
    - Confirmation question: Did you mean close the door?
    - Reprise clarification question: What needs to be closed behind the vehicle?

What Types of Questions are Most Frequent?

- For sample input Make sure you close the XXX behind the vehicle
  - Continue without asking a question (infer XXX or inference unnecessary) ***62%***
  - Stop and ask a question ***38%***
    - Generic question: What did you say? ***6.2%***
    - Confirmation question: Did you mean close the door? ***2.6%***
    - Reprise clarification question: What needs to be closed behind the vehicle? ***27.7%***

Question Types and Guessed POS



Implications and Future Work

- In 2/3 of cases, Turkers felt they did not need to ask a question
- In ~3/4 of cases when Turkers chose to ask a question, it was a targeted (reprise) clarification question
  - People prefer to ask targeted clarification questions, especially for missing content words
  - Hard to create reprise questions when missing word a wh-word or preposition
    - But.. could infer missing word when it was a function word or action verb
    - Didn’t ask questions

Can SDS Be Taught to Do the Same?

- Decide whether to infer the missing word and continue, or ask a Reprise Clarification Question
- What does this require?
  - Identifying ASR error locations within an utterance precisely
  - Inferring part-of-speech of misrecognized word
  - Hypothesize real word or compose appropriate clarification question to elicit a correction from the user

Outline

- Building a Dialogue Manager for Speech 2 Speech Translation
- Data Collection for Clarification Questions
- Classification experiments
  - Predicting user behavior
  - Identifying local errors
  - Predicting error type
- Future research

Two Experiments: Continue? Reprise?

- Goal: Predict whether a person will infer a word and continue or stop to ask a question
- Method:
  - If majority of Turkers chose to ask a question, label the misrecognized utterance ‘stop’, o.w. not
  - If at least one Turker decided to ask a Reprise question, label it ‘reprise’, o.w. not

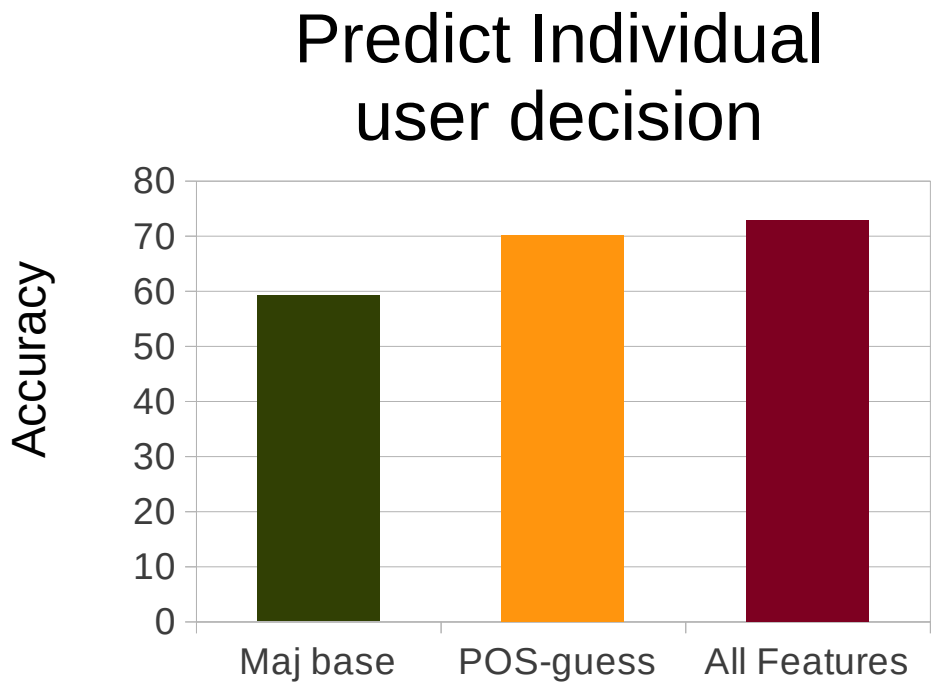
Features Used in Classification

- Error word position (first, last, middle)
- Part of Speech
  - Automatic (Stanford tagger on transcript)
  - User's guess
  - POS n-gram
  - All bigrams and trigrams of POS tags in sentence
- Syntactic dependency
  - Dependency tag of misrecognized word
  - POS tag of the syntactic parent of the misrecognized word
- ●

- Semantic role (Senna SRL parser)
  - Label of the error word
  - All semantic roles present in a sentence

Stop/Continue Experiment

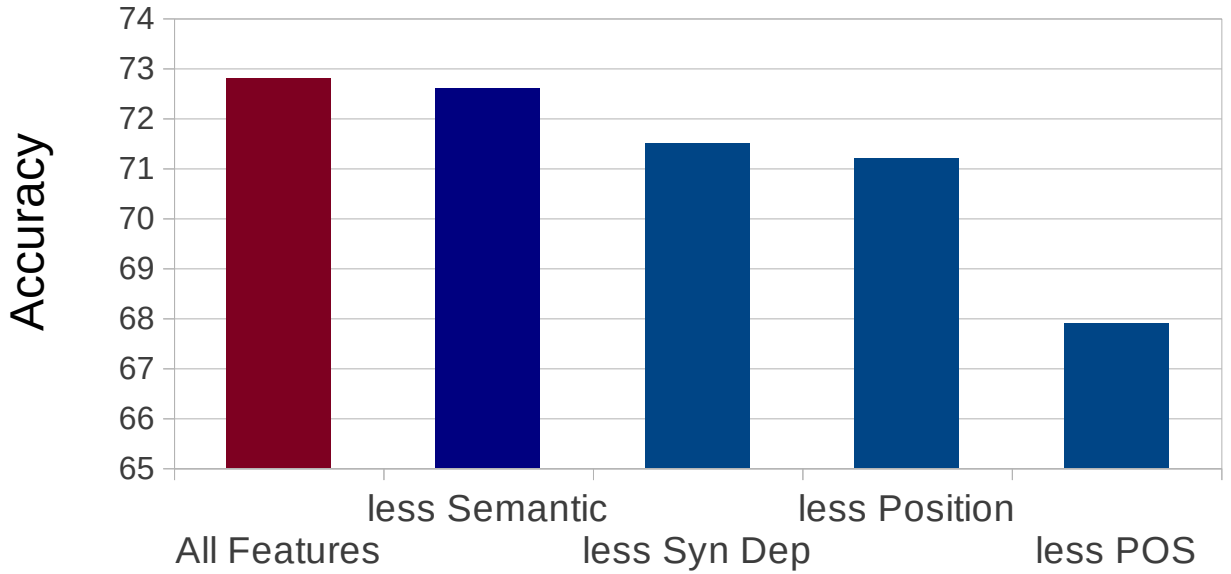
Predict if a user stops to ask a question or continues
Ignoring the error? 13.7% improvement over baseline



Machine learning using Weka with C 4.5 decision tree

Stop/Continue Experiment

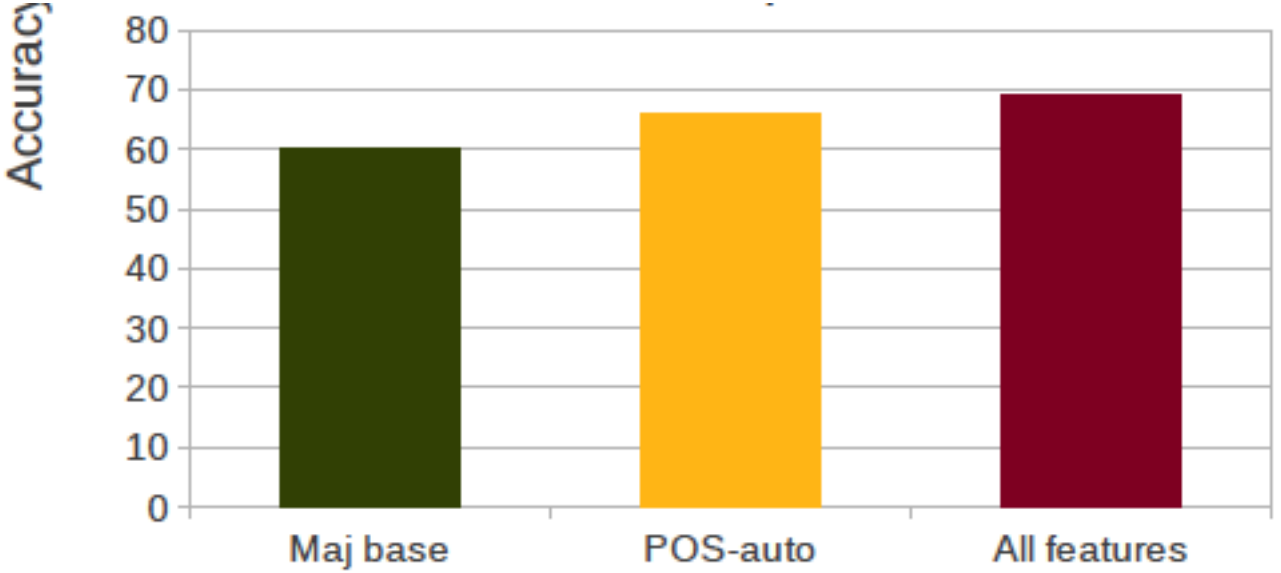
POS features most important



Machine learning using Weka with C 4.5 decision tree

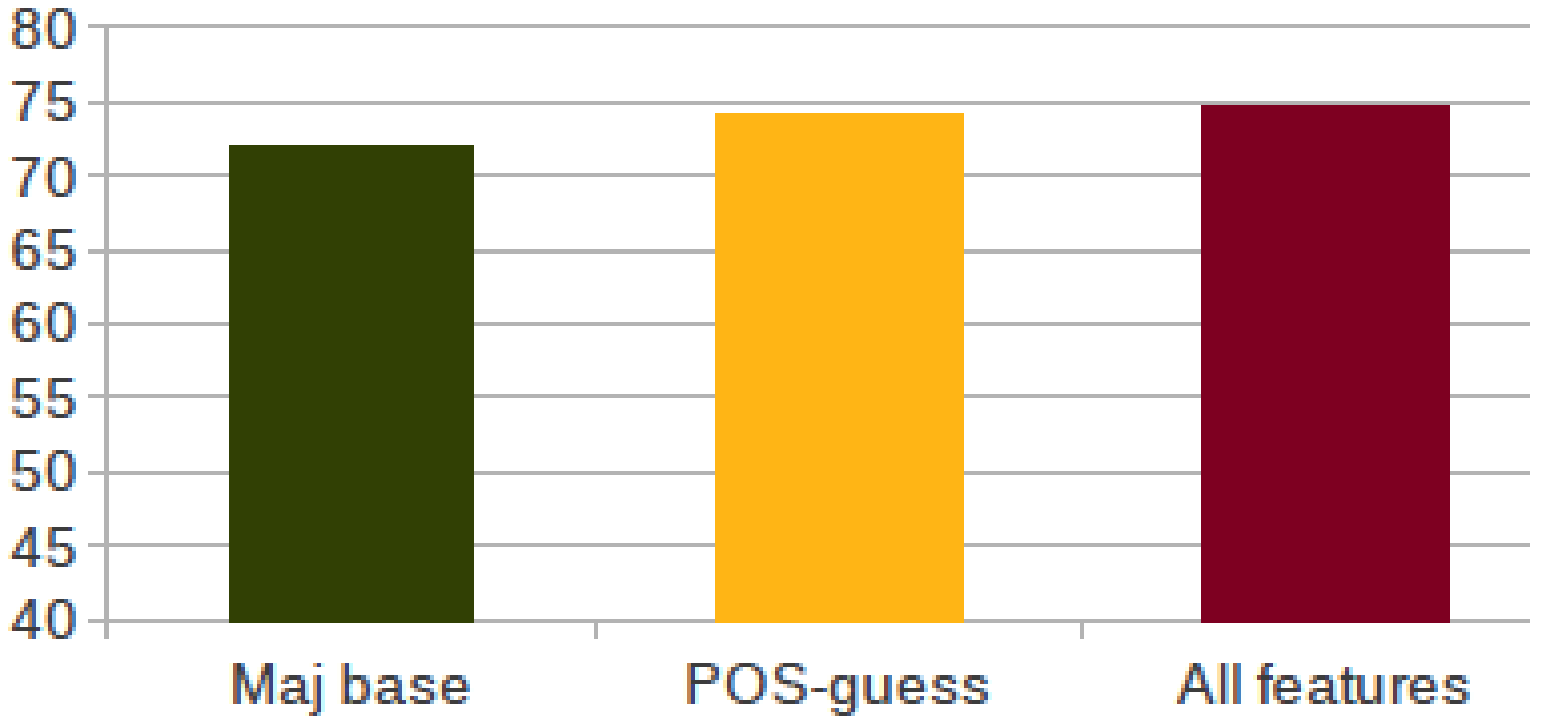
Predict Collective User Decision to Stop or Continue

- Decision = 'stop' if at least two annotator chose to stop
- Improve accuracy by 9.6% over baseline



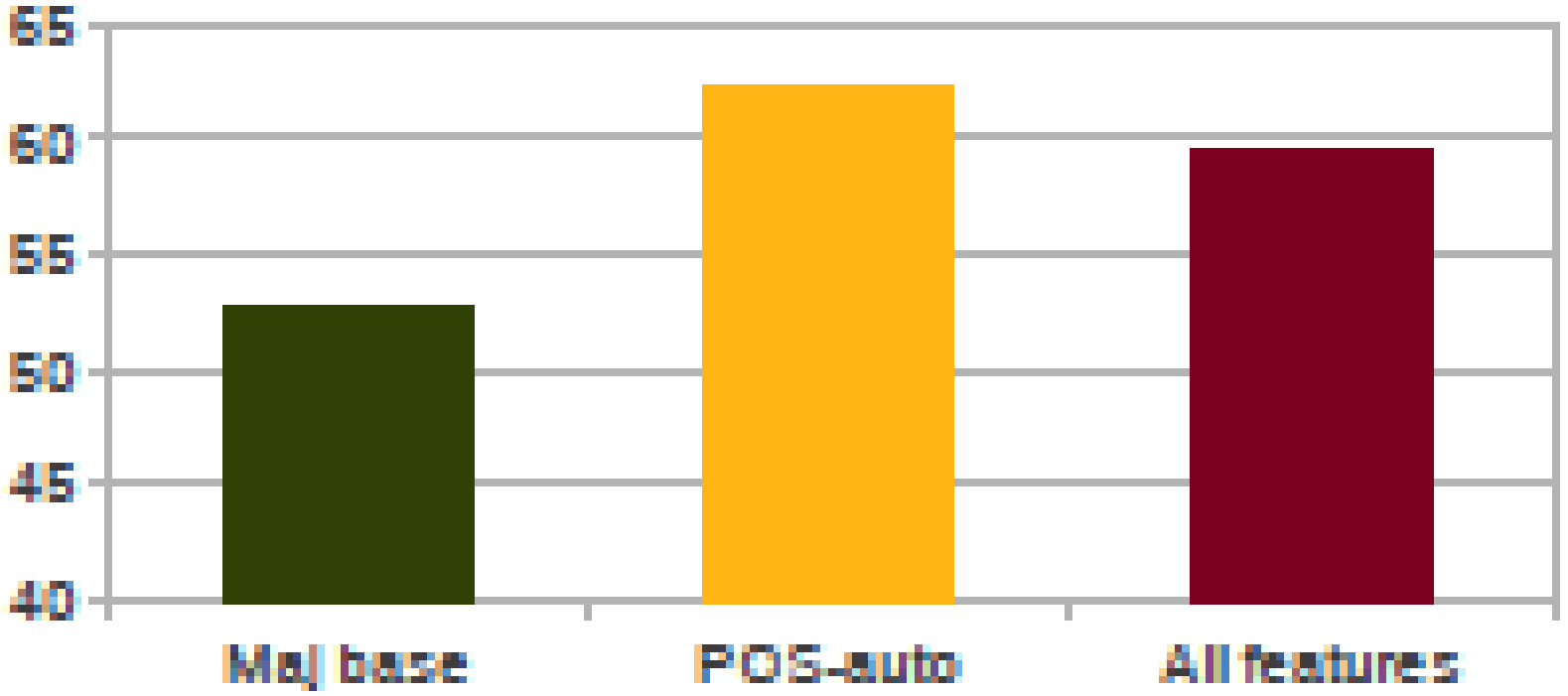
Predict whether possible to ask a Reprise Question: Individual Decisions

- All features increase accuracy by 2.1% points over baseline



Predict whether possible to ask a Reprise Question: Collective Decision

- POS increases accuracy by 9.7% points over baseline



Outline

- Building a Dialogue Manager for Speech 2 Speech Translation
- Data Collection for Clarification Questions
- Classification experiments
  - Predicting user behavior
  - Identifying local errors
  - Predicting error type
- Future research

Localized Error Detection

- ***Goal:***
  - Tokenize ASR hypothesis into correctly recognized segment(s) and incorrectly recognized segment(s) based on features derived from the hypotheses.
  - Use correctly recognized segments to generate a targeted clarification question.
- Machine learning experiments to determine an optimal feature set for performing localized error detection.
  - Word level
  - Utterance level

Utterance Level Features

- Baseline: Avg ASR confidence score for all words in utterance
- Optimal Predicators:
  - Avg ASR conf score for all words in utt
  - Average word-length in utterance
  - Utterance length in words
  - Utterance location within corpus
  - POS unigram & bigram count
  - Ratio of func words to total words in utt

Word Level Features

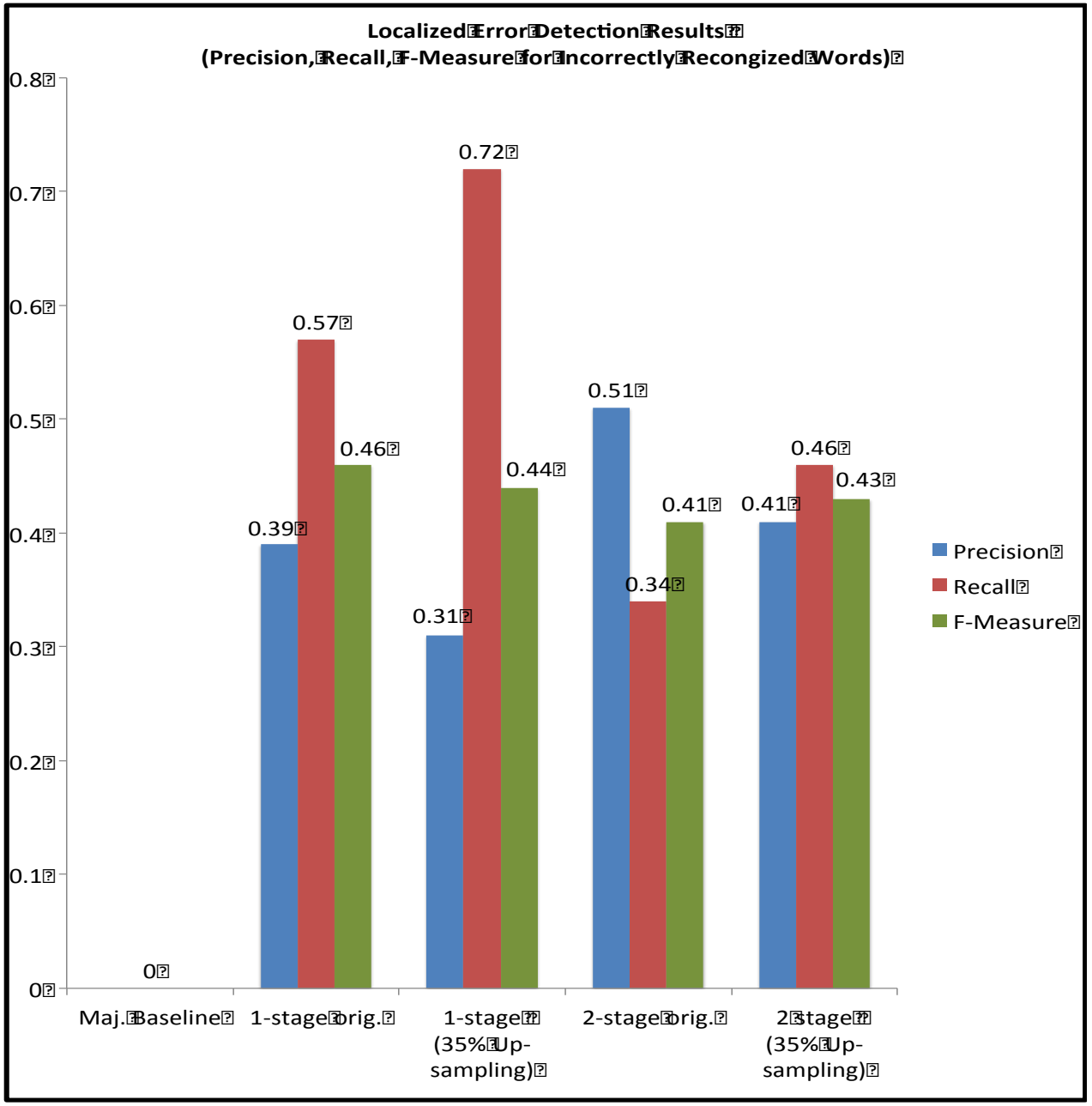
- Baseline: ASR Confidence Score
- Optimal Features:
  - ASR conf score for current word
  - Avg ASR conf score for current word & current word context
  - Avg ASR conf score for all words in utt
  - Word length in letters
  - Max-length word frequency in utt
  - Utterance length in words
  - Utterance location within corpus
  - Word distance from start of sentence
  - POS tag (curr, prev, next)
  - Func/Content tag (curr, prev, next)
  - Ratio of func words to total words in utt

Non-Optimal Features

- Information associated with minimum-length word In utterance
- Fraction of words in utt with greater length than avg-length word in utt
- Syntactic features such as dependency tag of current word
- Prosodic features such as jitter, shimmer, pitch, and phrase information
- Semantic information obtained from a semantic role labeling of data

Experiments

- To simulate actual performance we conduct 1-stage and 2-stage experiments with and without up-sampling
  - 1-stage: Classify each word in the corpus
    - The 1-stage (with 35% up-sampling) approach yields the highest recall for detection of word mis-recognition at 72%.
  - 2-stage: First classify all utterances as correct or incorrect, and then only classify the words in the utterances classified as incorrect
    - The 2-stage (no up-sampling) approach yields the highest precision for detection of word mis-recognition at 51%.


Localized Error Detection Results
(Precision, Recall, F-Measure for Incorrectly Recongized Words)
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.39
0.57
0.46
0.31
0.72
0.44
0.51
0.34
0.41
0.41
0.46
0.43
Maj. Baseline
1-stage orig.
1-stage (35% Up-sampling)
2-stage orig.
2 stage (35% Up-sampling)
Precision
Recall
F-Measure

Predicting Error Type

- What is the POS of the misrecognized word?
  - Is it a function word or a content word?
  - If a content word, is it an action verb?
- Motivation:
  - Automatically correct utterances with misrecognized function words or action verbs
  - Otherwise, ask a targeted clarification question
- Classification experiments on preposition detection (f=.72) and correction (f=.42): 24% and 68% over simple bigram baselines

Summary

- Improving communication in Spoken Dialogue Systems
  - Collecting data on when and how humans seek clarification to build SDS that can do the same
  - Discovering features that can predict user behavior
  - Localizing likely ASR errors
  - Classifying error types, to enable SDS to know when to ask for clarification

Future Directions

- Can we ***automatically detect and correct*** simple errors such as function words or action verbs?
- Can we distinguish ***user reaction to appropriate vs. inappropriate questions*** automatically?
- How can an SDS decide to ***stop trying to clarify*** and allow the user to start over or move on?

Acknowledgments

- Svetlana Stoyanchev: AT&T Labs Research
- Sunil Khanal, Alex Liu, Ananta Pandey, Eli Pincus, Rose Sloan, Mei-Vern Then, Jingbo Yang: Columbia University
- Philipp Salletmayer: Graz University of Technology

Thank you!