Natural Language Understanding and Prediction Technologies

Nicolae Duta

Cloud ML @ Microsoft

Outline

- Voice and language technologies: history, examples and technological challenges
- Short intro to ASR: modeling, architecture, analytics
- Language prediction (aka modeling)
- Natural Language Understanding
  - Supervised learning approaches: training & annotation issues
  - Semi-supervised learning approaches
  - Parsers & hybrid models, multilingual models
  - Client-server architectures, dialog & semantic equations
- Human interaction with voice & language technologies
- Semantic web-search
- Disclosure

Deployed language technologies

Most applications that translate some signal into text employ a Bayesian approach:

$$\arg\max_{\text{sentence}} P(\text{sentence} \mid \text{signal}) =$$

$$\arg\max_{\text{sentence}} P(\text{signal} \mid \text{sentence}) \times P(\text{sentence})$$

Applications

- Speech recognition
- Handwriting recognition
- Spelling correction
- Optical character recognition
- Machine translation
- Word/sentence auto completion

Technologies based on voice input

- Technologies that use spoken input for requesting information, web navigation or command execution
  - DA systems: *Nuance* (bNuance+PhoneticSystems), *BBN/Nortel*, TellMe/Microsoft, Jingle, Google, AT&T, IBM (mid 1990s)
  - Dictation/speech to text systems: *Dragon* (mid1990s)
  - TV close captioning BBN/NHK (early 2000s)
  - Automated attendant & Call routing: AT&T, BBN, *Nuance*, IBM (early 2000s)
  - Form-filling directed dialog (flight reservations) (early 2000s)
  - Personal assistants/Full web search: *Siri*/Apple, *Dragon Go*, Google Voice, *Vlingo/SVoice*, *Microsoft Cortana* (from 2008)
  - Many dedicated systems:
    - TV control + music/video management: DragonTV, *Xbox one*
    - Online banking & Stock price search
    - Product reviews & FAQ search
    - Medical fact extraction from medical reports

Technologies based on voice input: history

- Architecture: Speech recognizer + NLU + Dialog manager
  - Older systems: centralized, deployed in the customer’s processing centers
  - New systems: client-server, server deployed in the manufacturer’s processing center, client app on user’s (mobile) device
- NLU approaches:
  - Handwritten grammar rules (top-down): STUDENT, ELIZA
  - Context independent grammars from training text: Tina (MIT)
  - Supervised text classification
  - Context-dependent parsing
  - Hybrid
- DARPA programs:
  - ATIS (Airline Travel Information System):1990-1994
  - Hub4 (Broadcast News LVCSR): 1995-1999
  - EARS (Broadcast News + Conversational LVCSR): 2002-2005
  - Gale (Speech to speech translation): 2005-2010

Comparing voice & language input

- Y. Wang, D. Yu, Y. Ju & A. Acero: “An introduction to voice search”, IEEE Signal processing magazine, May 2008

	User input utterances		Target semantic representation	
	Query naturalness	Input space	Semantic resolution	Semantic space
DA	**Low**	**Large**	**Low**	**Small**
Call routing	**High**	**Medium**	**Low**	**Small**
Directed dialog	**Low**	**Small**	**Low**	**Small**
Mixed-initiative dialog	**Low-Medium**	**Small**	**High**	**Small**
Voice search	**Medium-High**	**Large**	**High**	**Large**

Technological challenges

- Speech recognition: noise, very large vocabulary/OOVs, pronunciations
  - Noise: Environment noise or channel noise
  - Pronunciation: many foreign names, pronounced differently than in the native language
  - Speaker adaptation in most modern systems
- NLU: large semantic space, linguistic variation, recognition errors
  - The semantic entity distribution is skewed
  - Semantic entities come from noisy databases
  - Recognition errors: approximate matching
  - Hard to come up with a unified confidence measure in mixed systems
  - People may not use the “official” name of a concept
    - Generative methods: Generate possible ways of asking
    - Accepting methods: Incomplete parsing + guessing rules (users voluntarily provide category information “music by ****”)

Technological challenges

- Dialog management
  - Dialog turn dependent LM/NLU (tuned to the expected information type)
  - Explicit vs Implicit DMs
- Disambiguation: By additional cues like Location
- Tuning/Feedback: Is it possible to automatically learn from the user actions?

Brief Introduction to Automated Speech Recognition

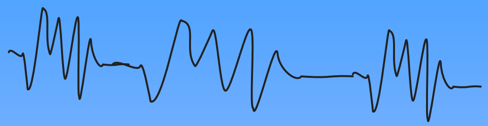
ASRs Bayesian approach:

$$\arg\max_{\text{sentence}} P(\text{sentence} \mid \text{audio signal}) =$$

$$\underbrace{\boxed{\arg\max_{\text{sentence}} P(\text{audio signal} \mid \text{sentence})}}_{\text{Acoustic score}} \times \underbrace{\boxed{P(\text{sentence})}}_{\text{Language score}}$$

Acoustic modeling (classical approach)

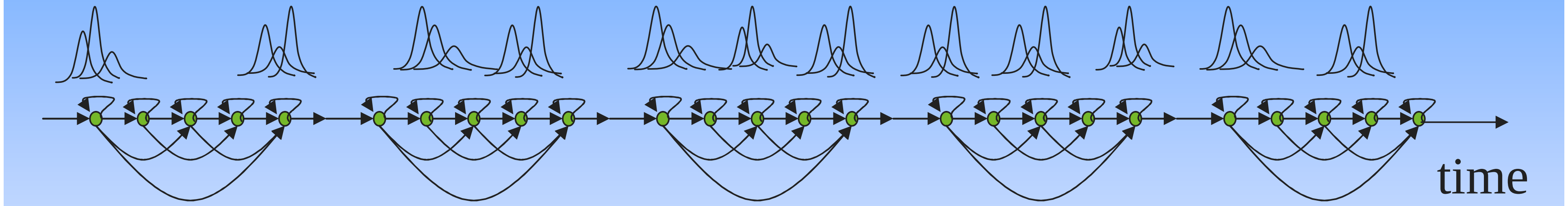
Signal pattern



Label string

[&][th][e][d][o][g][&]

Model



time

(&)/th/(e) (th)/e/(d) (e)/d/(o) (d)/o/(g) (o)/g/(&)

5-state left-to-right hidden Markov model with GMM state distribution

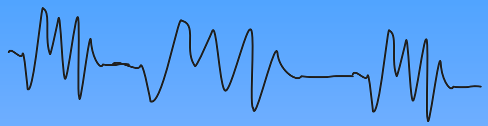
Model parameters

- Transition matrices
- Probability density function (pdf) for each state (mean vectors, covariance matrices and mixture weights for gaussian mixture models)

- The randomness in the state transitions accounts for time stretching in the phoneme: short, long, hurried pronunciations
- The randomness in the observations accounts for the variability in pronunciations

Training issues: acoustic models

- Training data is NOT manually segmented into phonemes



the dog [&][th][e][d][o][g][&]

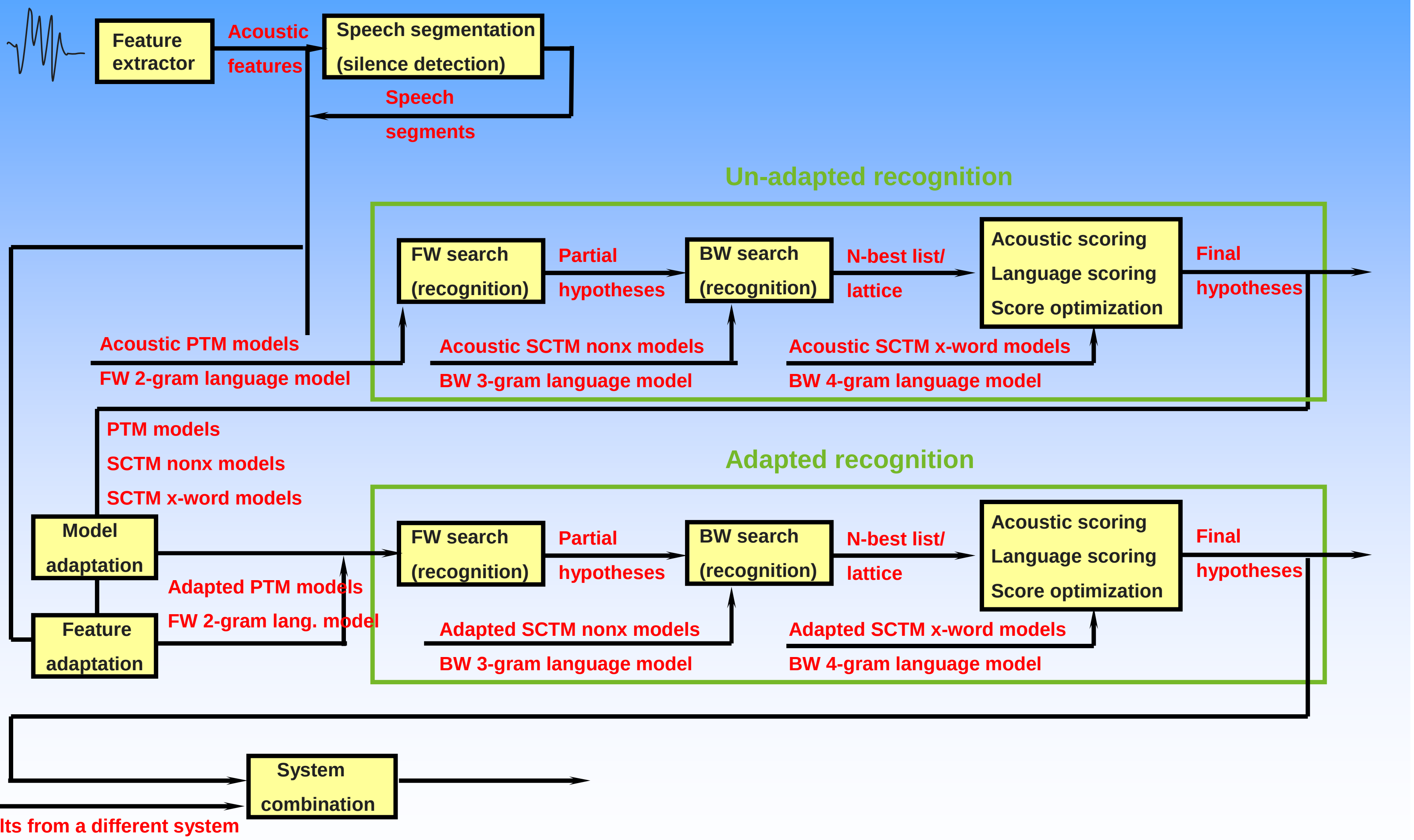
- The co-articulation effect: phonemes depend on their neighbors (context) ⟹ phoneme models in a triphone context ("triphone models") ⟹ ~(50^3) models each with a Gaussian mixture and state transition matrix
- Data available not sufficient to estimate all parameters ~(10^7)

  ⟹ share them among the triphone models: tying

  - Phoneme Tied Mixtures (PTM): All triphone models belonging to the same phoneme share the same Gaussian means and variances, but not mixture weights. This reduces the number of mixtures from ~100,000 to ~50 (each with 256 Gaussians, for example))
  - State-Clustered Tied Mixtures (SCTM): Clusters of states (may be from different phonemes) share the same Gaussian means and variances (but not the weights). The number of mixtures is reduced to ~2000 (each with 40 Gaussians)
  - Tied Mixtures: All the triphones share the same Gaussian means and variances (1 mixture with about 10,000 Gaussians)

  Tied Mixture Weights

Large vocabulary continuous speech recognition: the BBN EARS system



ASR Analytics

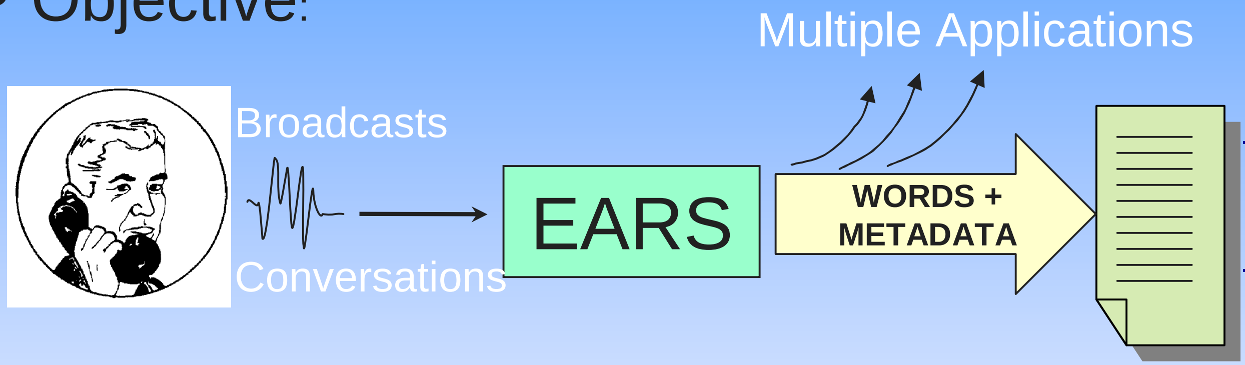
- Word Error Rate (WER): quality of the output produced by a speech recognizer
  - Measured against a human-made ground truth reference of the audio input
- Error types
  - Word substitutions
  - Word deletions
  - Word insertions

$$WER = \frac{Sub + Del + Ins}{\# \, refwords}$$

- WER varies a lot across the population
  - Smaller for native people, men
  - Report percentage of the population for which $WER < X\%$

The DARPA EARS program

- EARS: Effective, Affordable, Reusable, Speech-to-Text
  - DARPA program, funded sites: BBN/LIMSI, SRI, Univ. of Cambridge
- Objective:

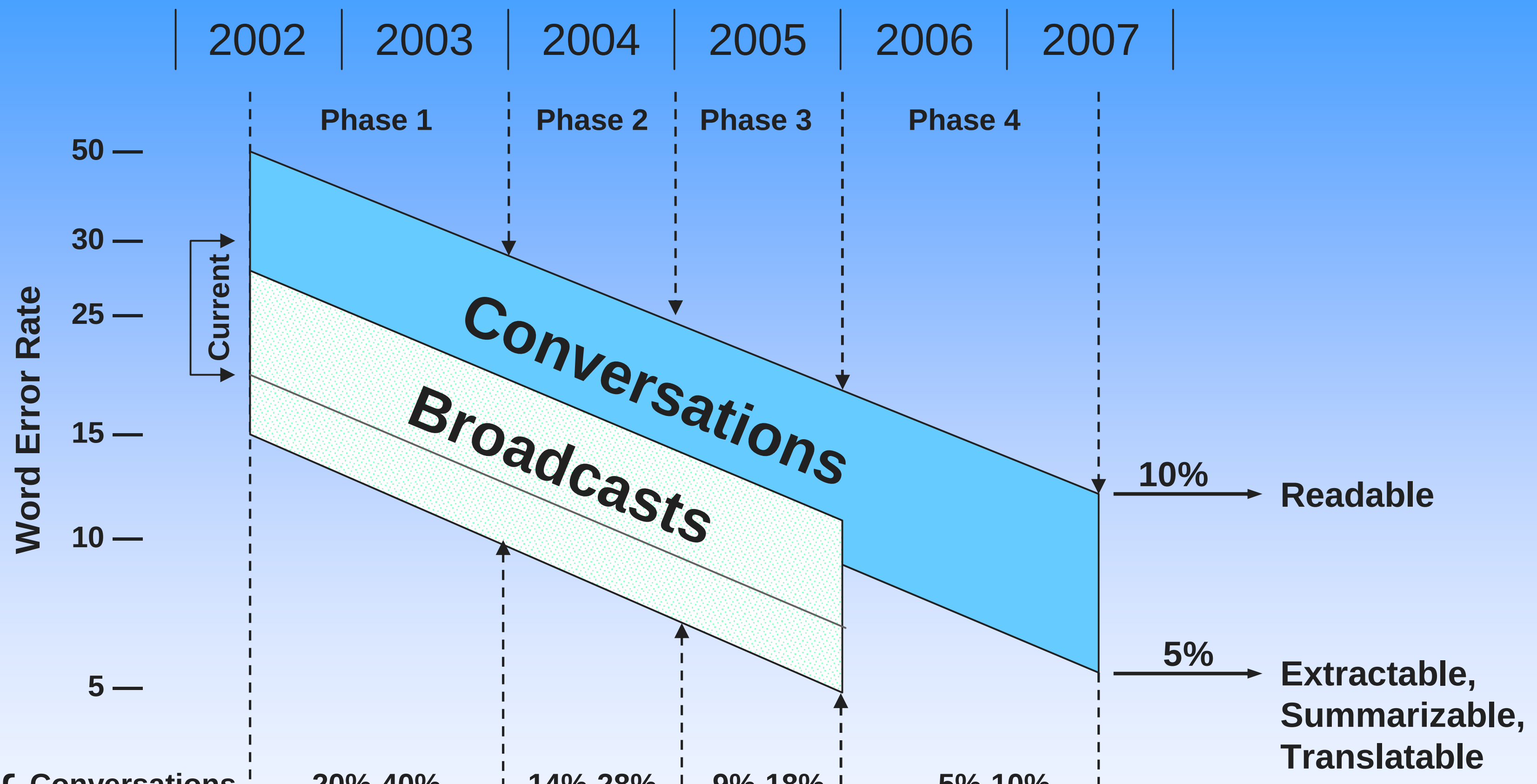


Accurate enough for

- Humans to read and understand easily
- Machines to detect, extract, summarize, translate

- **Program goals and evaluations**
  - Speech-to-Text (STT) 27.5% reduction in word error rate per year
  - Two Conditions: Conversational Telephone Speech (CTS), Broadcast News (BN)
  - Three Languages: English, Arabic, Chinese
- Tests
  - Annual new test data
  - Progress tests (same every year)

EARS program performance targets



	Conversations	20%-40%	14%-28%	9%-18%	5%-10%
	Broadcasts	12%-24%	8%-16%	5%-10%	---
Actual	Conversations	18%	15%		
	Broadcasts	12%	10%		
Speed	Conversations	---	20X	10X	1X
	Broadcasts	20X	10X	1X	---

Error Analysis (broadcast news speech)

1. Substitution of short or/and similar words: 20-25%

Ref	Hyp
americans who STRUGGLE to understand	americans who STRUGGLED to understand
to understand *** israelis and palestinians	to understand THE israelis and palestinians
that accompanied them IT was quite an	that accompanied them THAT was quite an
airlines with THE background TO that	airlines with A background OF that

- The correct word is usually in the hypotheses list.
- However, the LM is of little help in such cases, it is difficult even for a human to guess the “right” choice based on a short history

2. Errors generated by proper names (persons or locations): 15%

- Very costly: each mis-recognized name word generates 1.5-2 errors. Longer names are split into more words (BRASWELL ➔ BROWN AS WELL)
- Some (<1/4) of them are OOV (IVANISEVITCH). OOV rate is only 0.35%
- Many due to spelling differences (HANSSEN ➔ HANSEN)
- For most of them we do not have sufficient LM training.

Error Analysis (conversational speech)

1. Errors due to disfluences (mispronunciations, bad grammar, hesitations, fillers, edits, etc)

Ref: is it is too much OF AN EASY OUT well IF things do not work

Hyp: is it is too much AND HE IS YEAH well ** things do not work

she had a **** HAUNTED HOUSE (%hesitation) there was a BELL that would ****** RING AT a certain

she had a HARD TO HAVE %hesitation there was a BELLOW that would BRING IT TO a certain

2. High deletion rate (ratio deletions/insertions = 2.5:1, for BN it is only 1.5:1)

- In most cases, the words are largely inaudible – should have been marked as un-intelligible

Ref: kind of strange for me to (%hesitation) YOU KNOW all my … and EVERY ONE IS HAVING babies

Hyp: kind of strange for me to *** **** all my … and ***** *** ** ****** babies

and maybe AT a lower stage of development maybe AT a higher stage OF DEVELOPMENT THAN we are

and maybe ** a lower stage of development maybe ** a higher stage ** *********** THAT we are

3. Long words which are misrecognized are split into several words

what * * * * a UNIVERSAL CARD from * BLOCKBUSTER'S	what ARE YOU IN OVER a *** CAR from PROBLEM BUSES
a big ****** ******* AMPITHEATER like	a big CAMPUS THEATER NOT like
you know ** ***** *** MINESWEEPER	you know MY SWEET FOR HIM
**** ** TALASSEE alabama	TELL US THE alabama

What is the ground truth in speech recognition?

- Manual transcriptions differ among people
  - Some of the error may be carelessness
  - Much of the speech is not audible
  - Much of it is true ambiguity
- CTS Eval03 was carefully transcribed by 6 different teams
  - There is an average 6% disagreement between any pairs of transcripts
  - Many times the transcribers produce “what the person should have said”
- We cannot expect to achieve WERs lower than the differences among transcribers

Tr1: and UH so THEN WHEN I you know i finally GET A CHANCE to go out with my husband it's
Tr2: and UM so THE ONLY TH- you know i finally *** DID ATTEMPT to go out with my husband it's

Language prediction (aka modeling)

Most applications that translate some signal into text employ a Bayesian approach:

$$\arg\max_{\text{sentence}} P(\text{sentence} \mid \text{signal}) =$$

$$\arg\max_{\text{sentence}} P(\text{signal} \mid \text{sentence}) \times P(\text{sentence})$$

If sentence $= w_1\ w_2 \ldots w_n$ and a two-word history is considered sufficient to predict the next word, then

$$P(\text{sentence}) = \prod_{i} \text{P}(w_i \mid w_{i-1}, w_{i-2})$$

Statistical Language Modeling

Objects (classes): 100K English words

Signal pattern: this is B B C

P(S) = P(C|B,B,is,this)*P(B|B,is,this)* P(B|is,this)*P(is|this)* P(this)

Assumption: history matters only up to a certain point

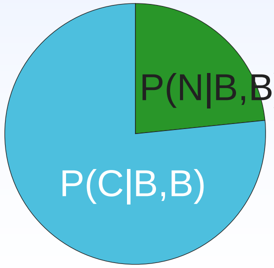
= P(C|B,B)*P(B|B,is)* P(B|is,this)*P(is|this)* P(this)

There are 10^{15} probabilities to estimate !

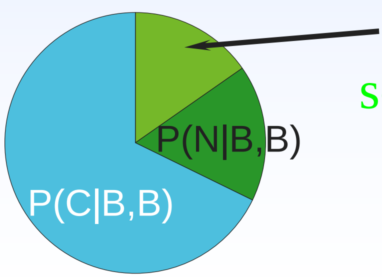
- The art of language modeling is dealing with sparse data, we usually do not have more than a few billion words of training so most word t-uples are unseen in training but we have to assign them probabilities
- Use discounting: set aside a part of the probability mass for the unseen target words

Training for history (B,B):

B B C, B B N



How likely is B B B?



Probability mass set aside for unseen targets

Approximating $P(w_i \mid w_{i-1}, w_{i-2})$

- Even with a large amount of training (1+ billion words) some 10%(BS) – 20%(CS) of triples (w_i, w_{i-1}, w_{i-2}) are not seen in training so their maximum likelihood probability is 0

- Apply interpolated discounting: set aside a part of the probability mass for unseen word sequences and recursively interpolate longer history probabilities with shorter history probabilities :

$$P_{ML}(w_i \mid w_{i-1}, w_{i-2}) * \alpha(w_i, w_{i-1}, w_{i-2}) + \beta(w_{i-1}, w_{i-2}) * P(w_i \mid w_{i-1})$$

- To maintain a probability model we need it to sum to 1 over w_i :

$$\beta(w_{i-1}, w_{i-2}) = 1 - \sum_{w_i} P_{ML}(w_i, w_{i-1}, w_{i-2}) * \alpha(w_i, w_{i-1}, w_{i-2})$$

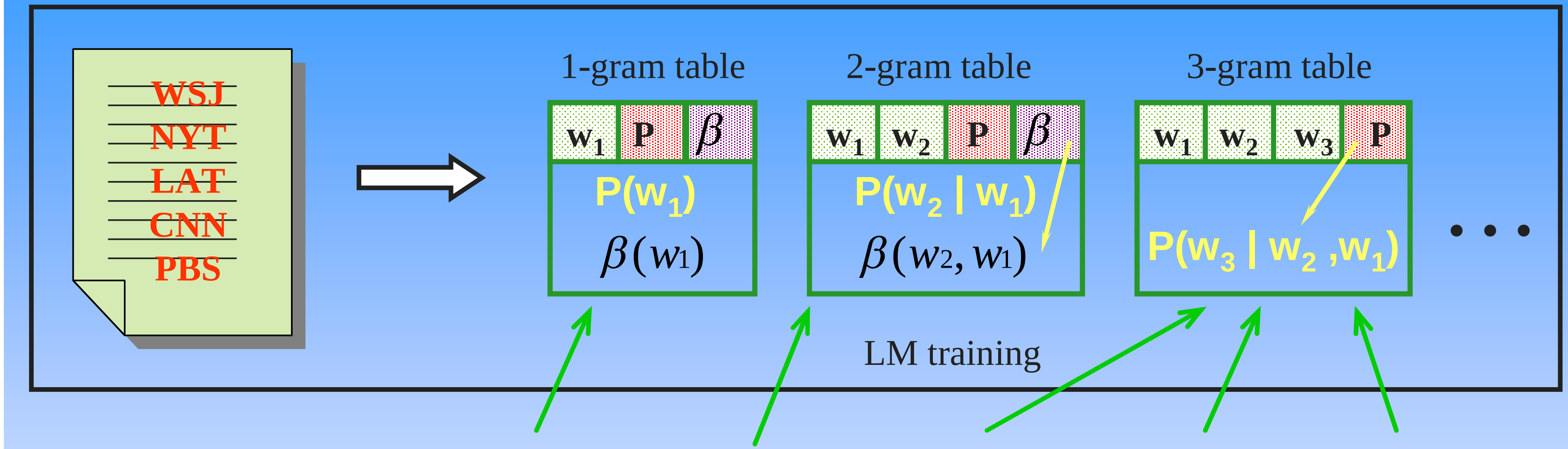
Witten-Bell

$$\begin{cases} \alpha = \dfrac{Total(. \mid w_{i-1}, w_{i-2})}{c * Uniq(. \mid w_{i-1}, w_{i-2}) + Total(. \mid w_{i-1}, w_{i-2})} \\ \beta = \dfrac{c * Uniq(. \mid w_{i-1}, w_{i-2})}{c * Uniq(. \mid w_{i-1}, w_{i-2}) + Total(. \mid w_{i-1}, w_{i-2})} \end{cases}$$

Knesser-Ney

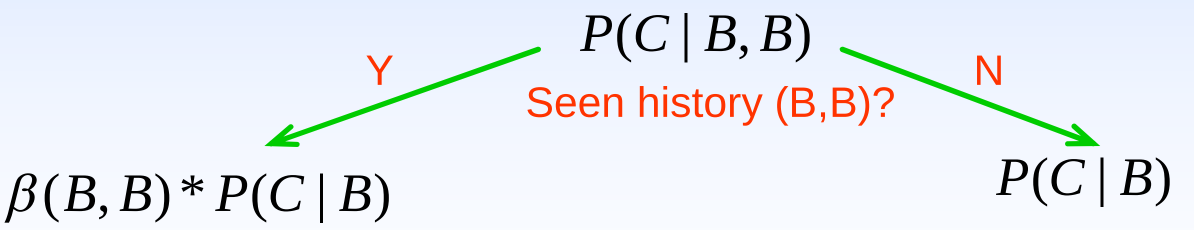
$$\begin{cases} \alpha = 1 - \dfrac{D(Count(w_i \mid w_{i-1}, w_{i-2}))}{Count(w_i \mid w_{i-1}, w_{i-2})} \\ \beta = \dfrac{\sum_{w_i} D(Count(w_i \mid w_{i-1}, w_{i-2}))}{Total(w_i \mid w_{i-1}, w_{i-2})} \end{cases}$$

How do we use a language model?



P(this is B B C) = P(this) * P(is|this) * P(B|is,this) * P(B|B,is) * P(C|B,B)

- What happens with n-grams for which we do not have LM training?
- If we have not seen (C B B) in training then:



$P(C \mid B, B)$

Seen history (B,B)?

Y

N

$\beta(B, B) * P(C \mid B)$

$P(C \mid B)$

- There is over a order of magnitude difference between a 3-gram probability and a 2-gram probability

Training a language model: size issues

- When we train an n-gram LM on a large corpus, most of the observed n-grams only occur a small number of times. There are 700M distinct 4-grams in a 1.5 billion-word corpus, more than half are only seen once!

- Due to computational constraints the singleton (seen only once) n-grams are usually discarded

- Questions:
  - Does the fact that an n-gram occurred one time provide useful information (is it statistically significant) ?
  - Is it practical to use a really large LM?

- The probability that a word is recognized is affected significantly by whether the corresponding n-gram is in the LM (measured by the "hit rate"), because if it is not, the LM probability (from backing off) is significantly lower.

Language model coverage

- English broadcast news test, (H4Dev03)
- Witten-Bell discounting with lower order smoothing

LM Order	Cutoffs [4g, 3g]	LM size [4g, 3g]	Hit Rates [4g,3g]	Perplex	WER
3	[inf, 6]	[0, 36M]	[0, 76%]	201	12.6
3	[inf, 0]	[0, 305M]	[0, 84%]	164	12.1
4	[6, 6]	[40M, 36M]	[49%,76%]	208	12.1
4	[0, 0]	[710M,305M]	[61%,84%]	139	11.8

- **Cutoff of 6 for trigram loses 0.5% absolute**
- **4-gram with cuttoff of 6 gains 0.5%**
- **4-gram cutoff of 6 loses 0.3%**

Mixing data from multiple sources

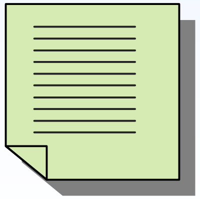
- Manual transcriptions of audio data
  - In-domain (current application)
  - Vertical domain (same industry)
  - General conversational data
- Automatic transcriptions of (in-domain) audio data
- Web-crawled text data
- Entries in large databases (census database)
- Human knowledge present in legacy hand built grammars

Mixing strategy: count or probability-based LM interpolation

Mining corpora for similar language patterns

- When we don’t have enough LM training data (CS: 5M words) we try to compensate by using out-of-domain (News: 1B words) data
- How do we know the out-of-domain data might be useful? We know it is not useful if it does not improve the overall “resemblance” of the training data to the test data (word t-uples present in both = hit rate)
- If the out-of-domain corpus is too large it is impractical to use or even compute a language model from all the data
- How do we select a News subset which is most relevant to (resembles best) the CS domain?
- News Mining: Use only those News sentences which contain a certain amount of word t-uples seen in the CS data

AND I I'M A BROADCAST JOURNALIST AND SO I FEEL LIKE ONE DAY I PROBABLY WILL
AND SO I I GUESS I'M I'M PRETTY EMOTIONAL ABOUT CRIME THINGS LIKE THAT NOW
YOU KNOW A LOT OF THINGS THAT HAPPEN THAT PEOPLE DON'T SEE



YOU DON'T SEE ANYTHING LIKE THAT NOW
I WOULDN'T DARE DO ANYTHING LIKE THAT NOW
YOU KNOW A LOT OF PEOPLE NEED PROFESSIONAL HELP
A LOT OF PEOPLE ARE BEING HURT BY ECONOMIC CHANGES
AND THEN I THINK EVENTUALLY I PROBABLY WILL
YOU CAN'T HOPE TO PROTECT INTELLECTUAL PROPERTY WITHOUT A TECHNOLOGY COMPONENT

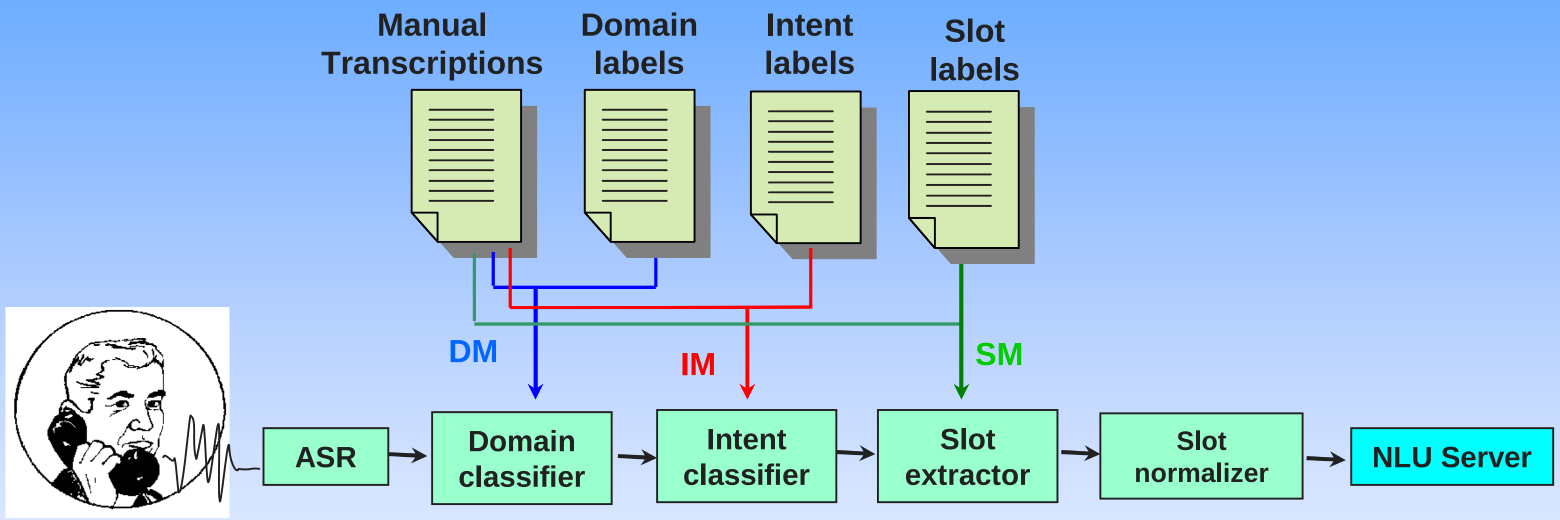
Discussion

- Even a single observed token of an n-gram tells you that it is *possible.*
  - It is important to know the difference between n-grams that are unobserved because they are rare and those that are impossible. [If we could really know this, we would have much better results.]
- The gain from keeping all n-grams is significant (0.5% for 3-grams, 0.3% for 4-grams).
- When using Knesser-Ney discounting the degradation is smaller, but there is still a loss
- However, when little training data is available the discounting method is very important and Knesser-Ney gives the best results

Natural Language Understanding

- Extracting meaning & information/metadata from text
- Applications
  - Personal assistants: command/transactions execution
  - Information retrieval / question answering
    - Direct questions: “who directed titanic”
    - Indirect questions: “find other movies by the director of titanic”
    - **Question understanding != Question answering**
  - Extracting structured information from unstructured text (eg, EHR)
  - Sentiment analysis
  - Automated recruiting (matching resumes to positions)
- Historical approaches
  - Knowledge / rule based
  - Statistical learning
    - Generative models
    - Discriminative models

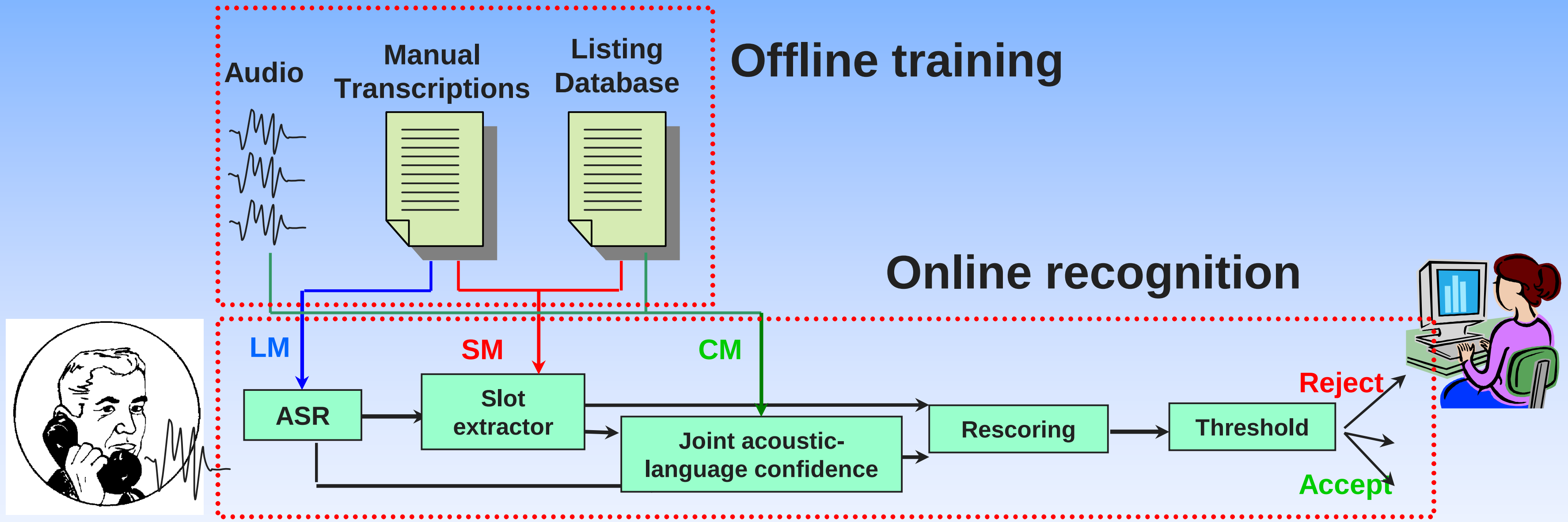
Supervised learning one-shot NLU architecture



- Top-down semantic modeling schema and data processing
- Domain/Intent classifiers are typically SVMs based on n-gram features
- One intent classifier for each domain, one slot extractor for each (domain,intent)

Directory assistance systems

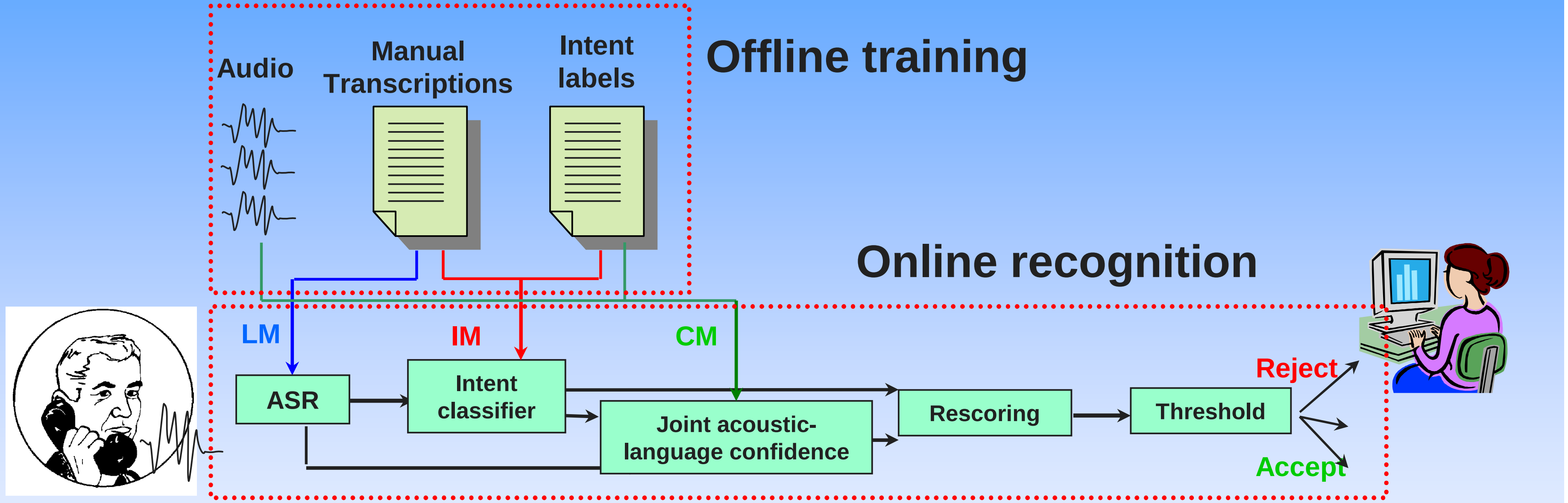
- Task: *automatically provide phone number/address for business & residential listings*



- State/large metropolitan area specific
  - Listing database contains (popularity) priors for businesses
  - Evaluation criterion: Automation rate

Call routing systems

- Task: *automatically route customer calls to the appropriate agent*

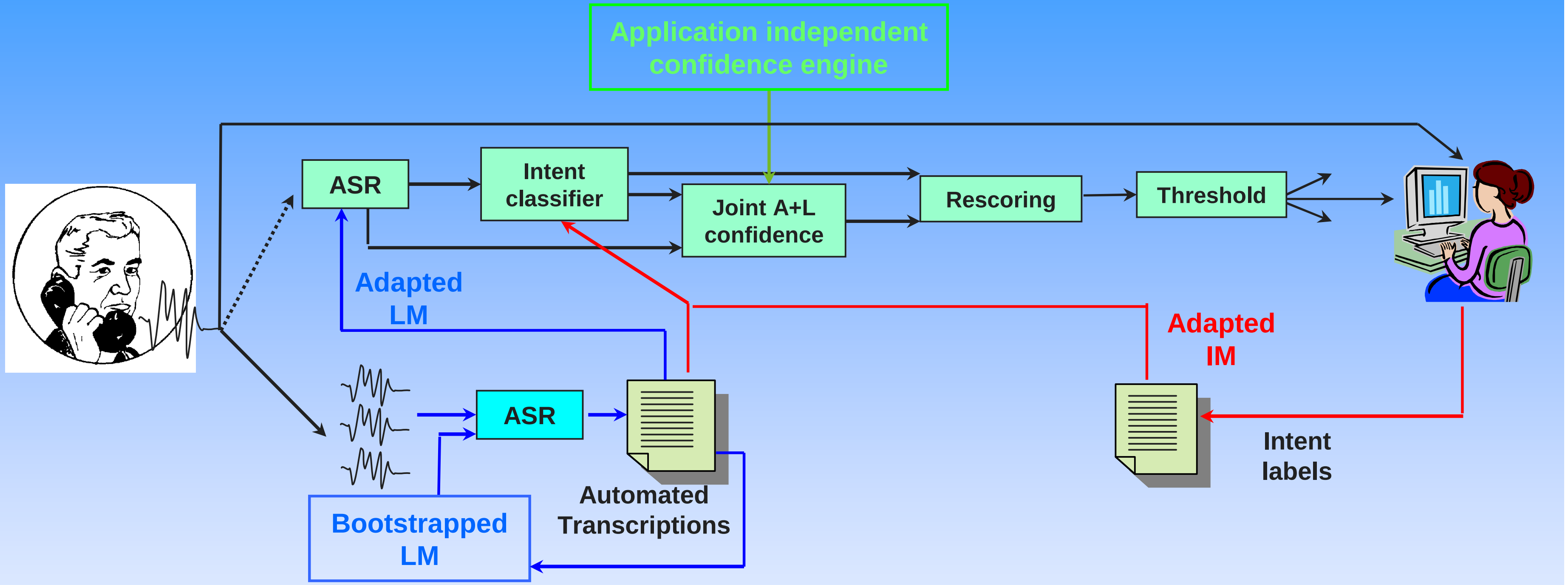


- Offline Learning
  - The application audio is manually transcribed and labeled before the system is deployed
  - The system may be retrained during tuning procedures

Design & training challenges

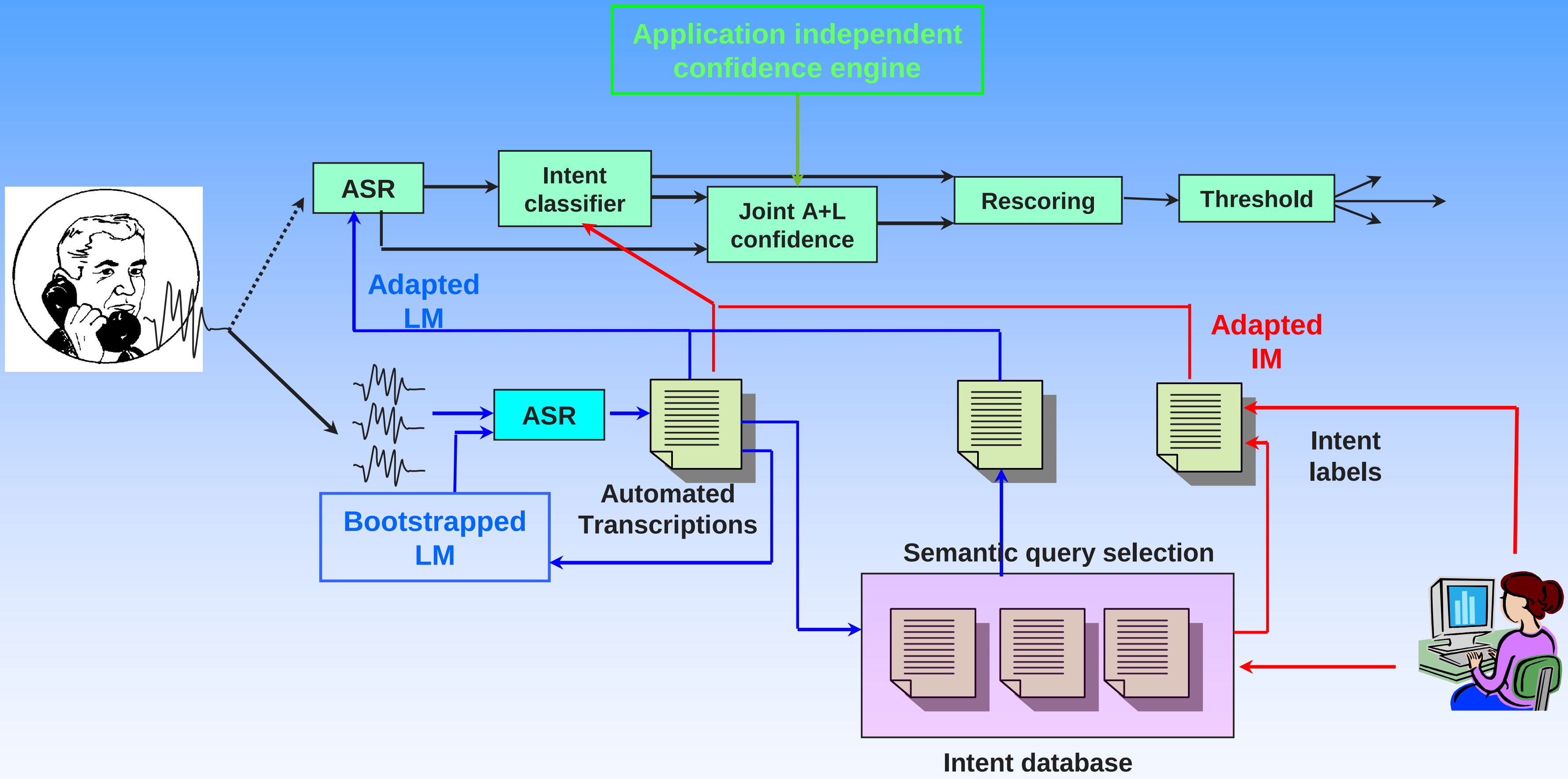
1. Requires large amounts of data annotation
   - Manual transcribing/labeling is costly, time consuming and tedious
   - Changes in the app spec/annotation schema requires data relabeling
2. Labeled data is usually inconsistent
   - Annotation schema may generate annotator confusion
   - Human annotators may be careless or cheating
   - Semantic labeling is very hard
3. Hard to explain & fix errors
4. Large number of models
   - Computationally intensive
   - High memory requirements
5. Semantic modeling schema is NOT based on data

Semi-supervised training



- Online Learning
  - The system operates in reduced automation mode while recording the incoming audio and operator actions
  - The system automatically transcribes audio and adapts its models whenever a given amount of new data has been collected

Semi-supervised training data selection



- Active Learning
  - The system queries a large query database for examples similar to the current task

Semantically-based query selection

- **Label the database sentences using all (well trained) intent classifiers available and produce a joint [intent, confidence]**
  - “speak spanish”: [SpanishApp/0.97; OOV/1; Unknown/0.96]
- **Label the query sentences using all classifiers available**
  - “espanol”: [SpanishApp/1; OOV/1; OOV/1]
- **For each query and database sentence, compute the posterior likelihood that they are assigned the same joint intent**
- **For each query, return the database sentences with the highest likelihood**
  - “espanol”: “is there anyone that talks in spanish”/0.00019677; “do you have someone who speaks spanish there”/0.00019574; “excuse me somebody speaks spanish”/0.0001909

Semantic ambiguity and confusion

- ***Confusable classes increase annotator inconsistency***

Customer request	***Semantic label (manual)***
need to talk to someone about my bill	*BillingAndPaymentsDis*
i'd like to talk to someone about my bill	*BillExplanation*
return d_s_l modem	*DisconnectDis*
d_s_l modem return	*InternetDis*
need r_m_a number	*DisconnectDis*
i need a r_m_a number to return my modem	*Sales*
due date	*BillExplanation*
payment due date	*AccountBalance*
i'd like to pay my bill later than the due date	*DisconnectDis*

Semantic ambiguity and confusion

- ***Personal assistant: Insert a new calendar event***

make an appointment with batman tomorrow
make an appointment with batman for tomorrow
create new appointment for wednesday at 3 pm
add an appointment at in four hours tuesday with james
add appointment for dentist at 7 pm on march 3rd
from 1 pm tomorrow i have a doctor's appointment with mike
i'm leaving for a doctor's appointment today at 11 am
set doctor appointment for april 12 2012 at 9 am
tomorrow doctor's appointment
susan the 16th' appointment 11 am

Legend: Title Invitee Date Time DontCare

Semantic ambiguity and confusion

Personal assistant: *Places domain*

open charlotte<absolute_location> restaurant<travel_header> page

show restaurants in mexico city<restaurant_location>

find best<restaurant_described_as> restaurants and hotels in rome italy <restaurant_location>

can you tell me more about the hotels<hotel_type> and restaurants in manila <hotel_location>

i am looking for hotel and restaurant<travel_header> information for manila <absolute_location>

display food and hotels in miami florida<restaurant_location>

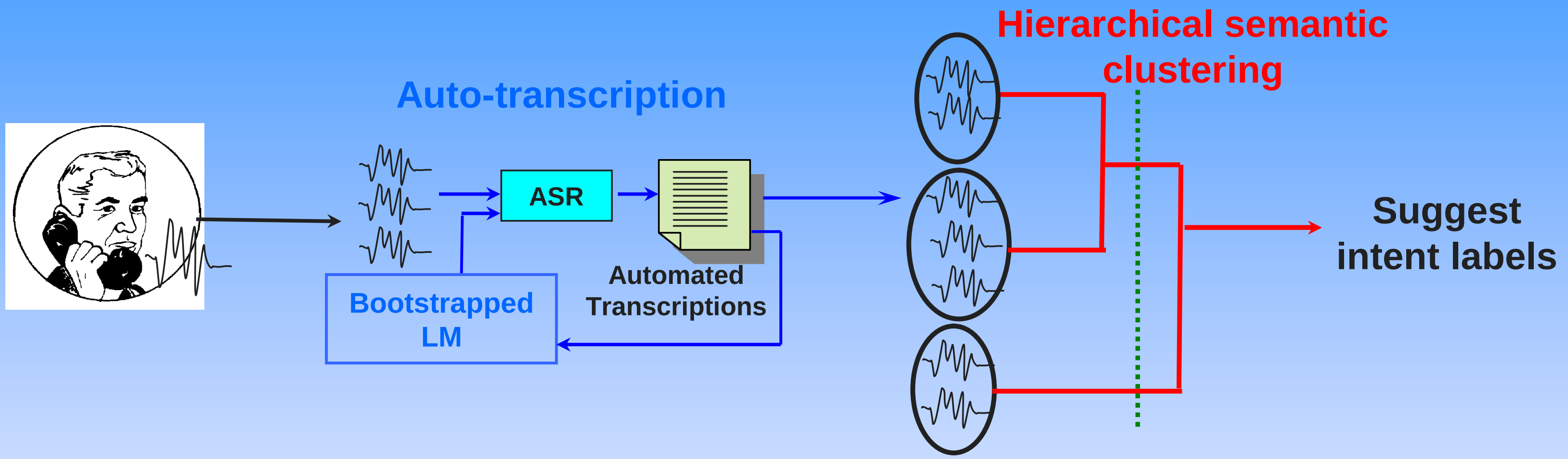
search for a starbucks<restaurant_name> near me<restaurant_near_ref>

find me a starbucks<restaurant_name> nearby<restaurant_near>

search nearby<local_biz_near> **walmart<local_biz_business_name>**

intersections<traffic_near> closest to evans<traffic_near> walmart <traffic_near>

Semantic clustering

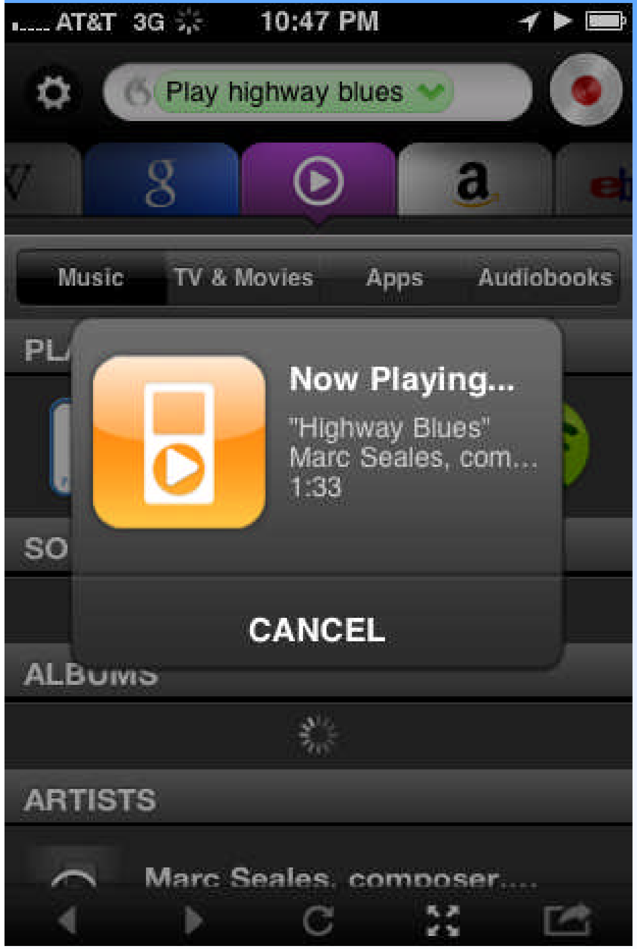


- Automatically suggest intent labels
  - Could one decrease the manual labeling time while increasing annotator consistency?

Voice search language understanding

- Query types
  - Navigational: reaching a website explicitly requested (e.g. “go to facebook”) or a certain state in the dialog flow (e.g. “go back” or ”cancel”),
  - Informational: finding information on the web (e.g. “capital grille restaurant reviews”)
  - Transactional: conducting a transaction on a website (e.g. “make a reservation at capital grille”)
- Short queries with high semantic resolution, large input space, increasingly in a natural language
- Manual annotations for supervised classifiers are very costly

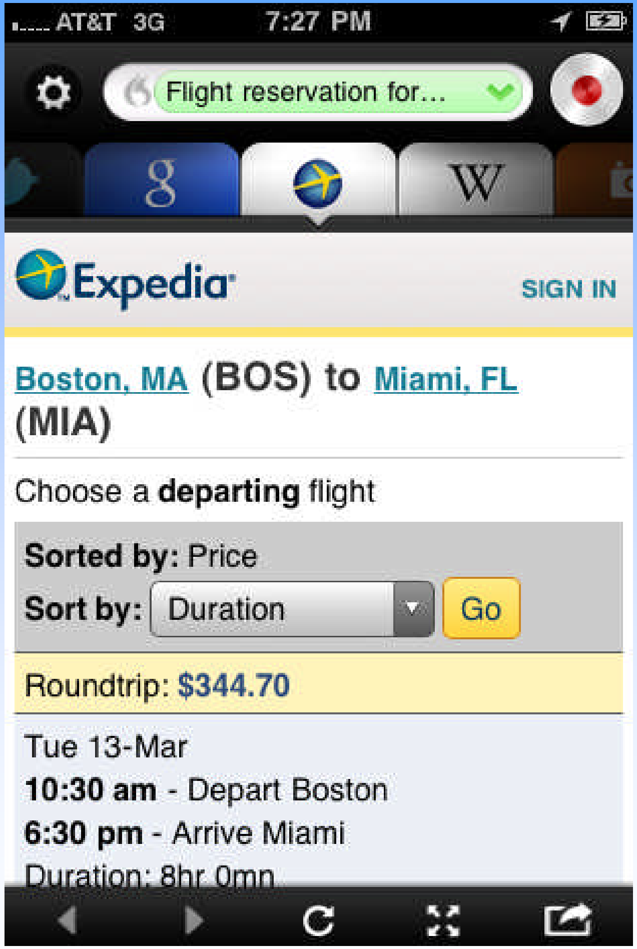
Voice-based personal assistant



Play highway blues

Now Playing...
"Highway Blues"
Marc Seales, com...
1:33

Marc Seales, composer....



Flight reservation for...

Boston, MA (BOS) to Miami, FL (MIA)

Choose a **departing** flight

Sorted by: Price

Sort by: Duration Go

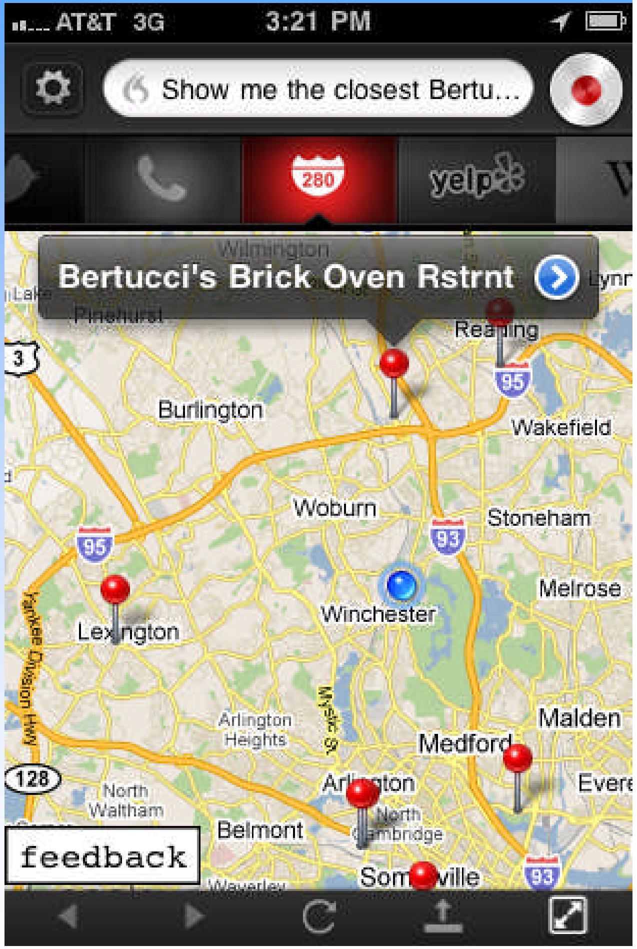
Roundtrip: **$344.70**

Tue 13-Mar

10:30 am - Depart Boston

6:30 pm - Arrive Miami

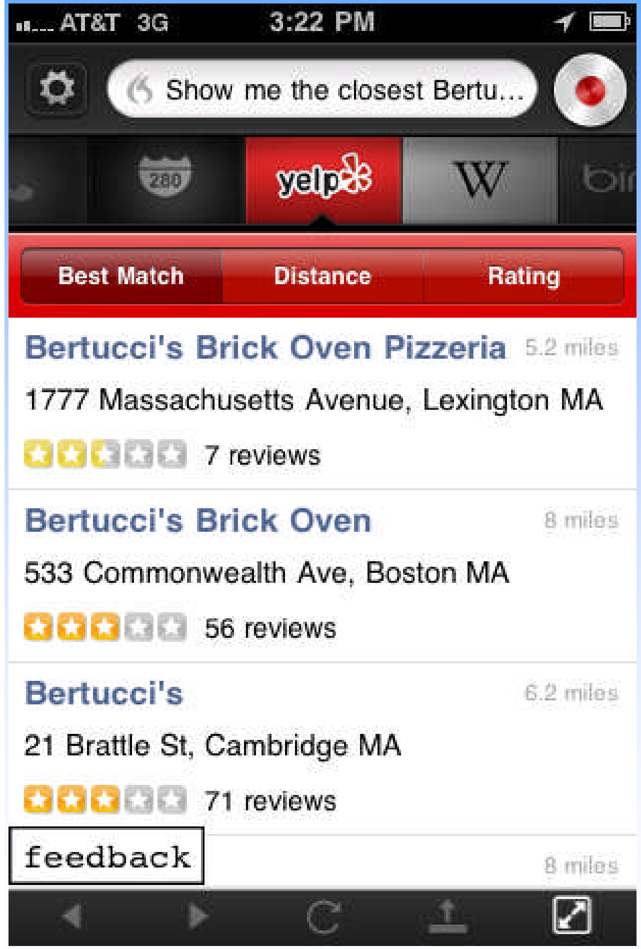
Duration: 8hr 0mn



Show me the closest Bertu...

Bertucci's Brick Oven Rstrnt

Voice-based personal assistant

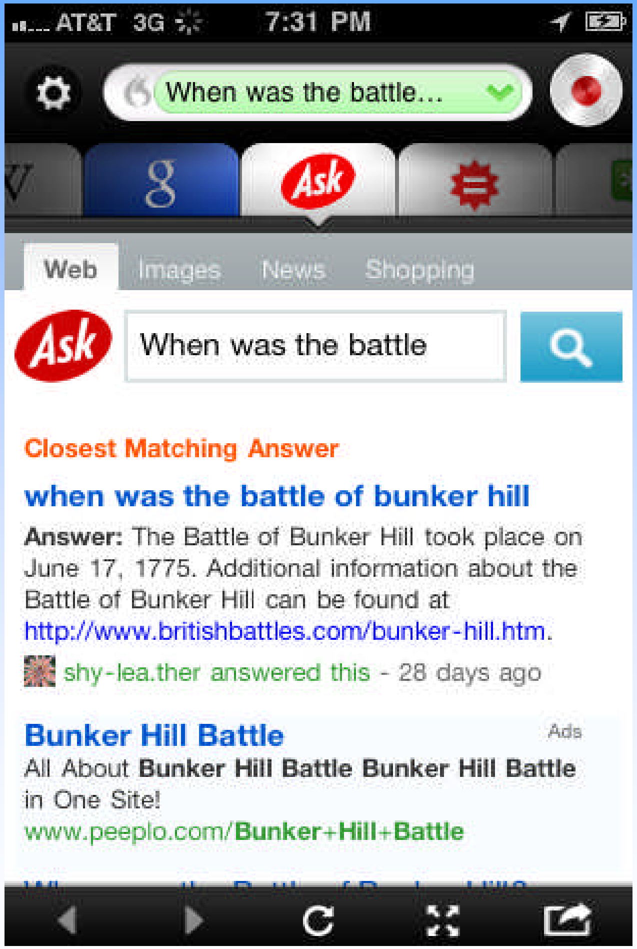


Show me the closest Bertu…

Bertucci's Brick Oven Pizzeria 5.2 miles
1777 Massachusetts Avenue, Lexington MA
7 reviews

Bertucci's Brick Oven 8 miles
533 Commonwealth Ave, Boston MA
56 reviews

Bertucci's 6.2 miles
21 Brattle St, Cambridge MA
71 reviews



When was the battle…

When was the battle

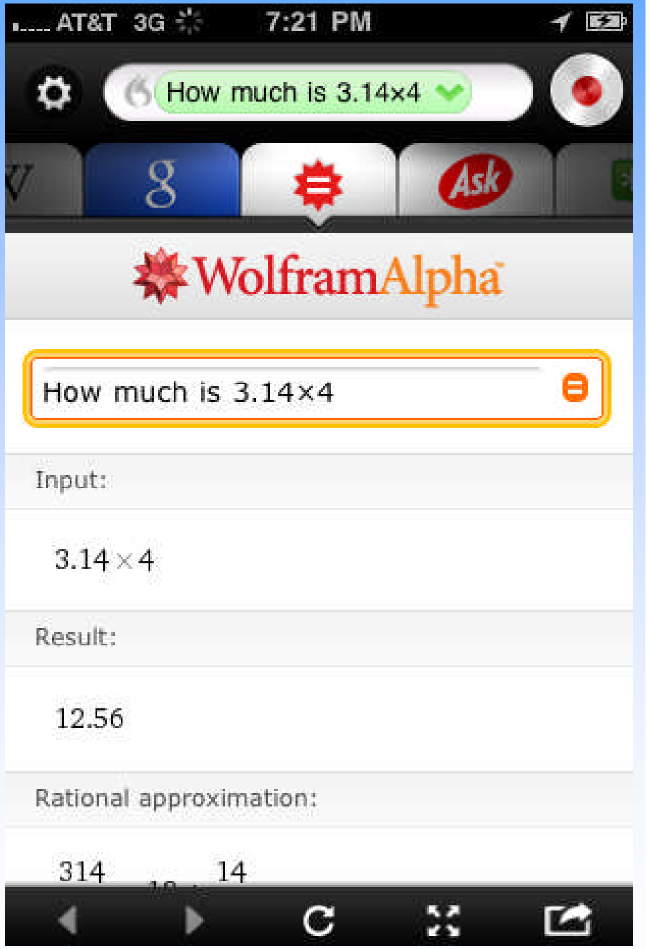
Closest Matching Answer

when was the battle of bunker hill

Answer: The Battle of Bunker Hill took place on June 17, 1775. Additional information about the Battle of Bunker Hill can be found at http://www.britishbattles.com/bunker-hill.htm.

shy-lea.ther answered this - 28 days ago

Bunker Hill Battle Ads
All About **Bunker Hill Battle Bunker Hill Battle** in One Site!
www.peeplo.com/**Bunker+Hill+Battle**



How much is 3.14×4

How much is 3.14×4

Input:

3.14 × 4

Result:

12.56

Rational approximation:

Semantic properties of voice queries

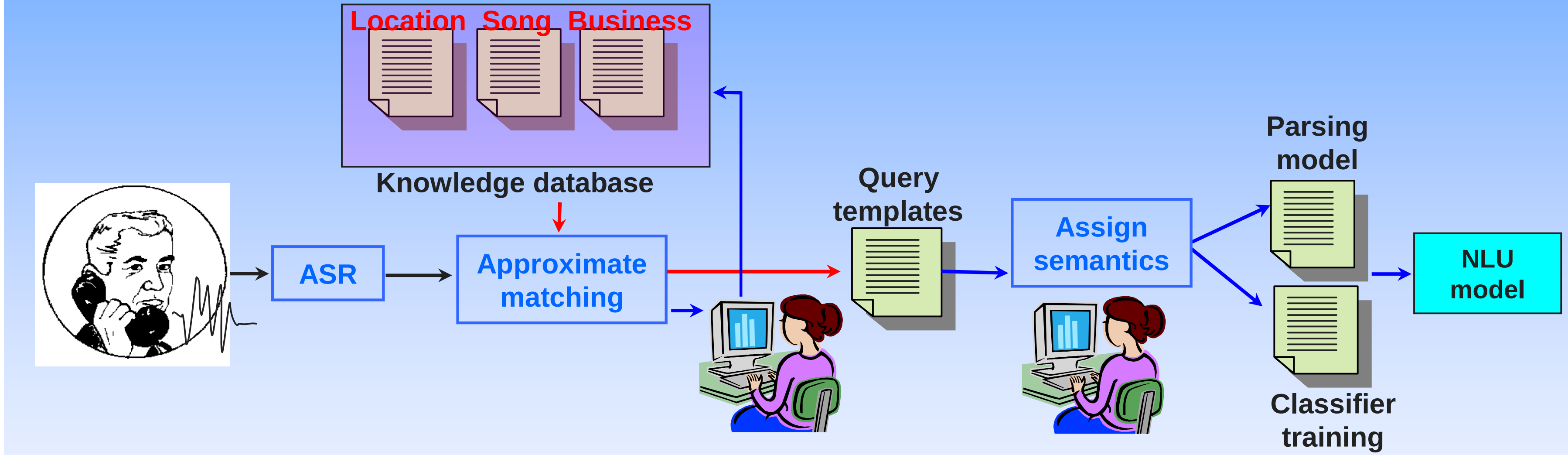
- **Most queries contain at least one name entity**
  - **Location**
  - **Person**
  - **Business**
  - **Media (Song/Album/Movie/Game/Show)**
- **Flat semantic structure: concatenation of an intent and one or more name entities**
  - **Intents: Search, Play, Buy, Call, Reserve**

Visualization strategy: Replace name entities / intent fragments by their semantic type

- **Compressed query = query template**

Semi-supervised query mining pipeline

- Discover the semantic structure (query templates) using approximate string matching
- Assign meaning (domain/intent/etc) to each query template
- Generate parsing rules and classifier training samples then train NLU classification models



Query	Query template	Domain	Query intent
wendy’s in boston	<Restaurant><Location>	Restaurant	find_FoodEstablishment
nearest kmart	<Near><Business>	Local Business	find_LocalBusiness
weather in 02155	<Weather><ZipCode>	Weather	check_weather
boston bruins scores	<SportsTeam><SportsInfo>	Sports	get_scores
details on call of duty	<Info><Game>	Games	show-details

Query template extraction

- Approximate string matching of gazetteer/dictionary items to the data

  "wal-mart frankfort kentucky",
  "walmart in daphne alabama" ⟶ ***<BUSINESS> <LOCATION>.***
  "apple store austin texas"

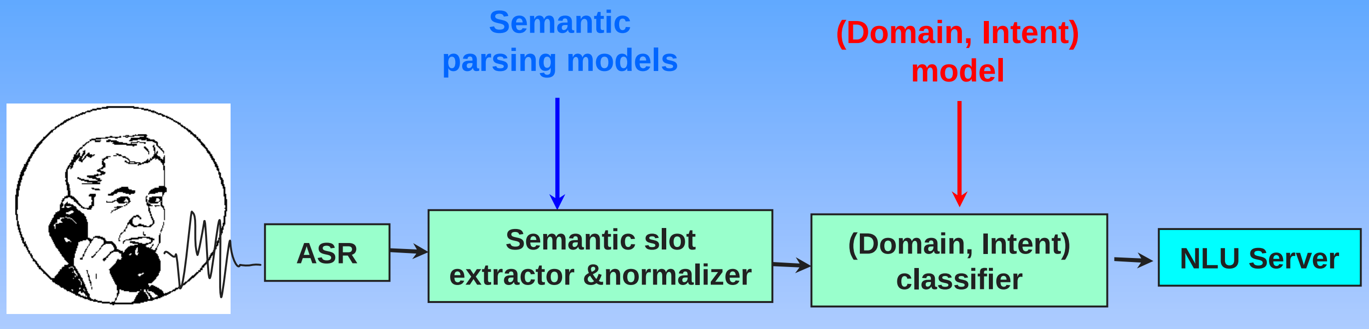
- A large number of specific queries can be abstracted into one query template
  - <BUSINESS> <LOCATION> covers 22818 queries

- The compression rate depends on:
  - The semantic domain
  - The semantic entity list
  - The contextual phrases which are modeled
  - The matching algorithm

Model training

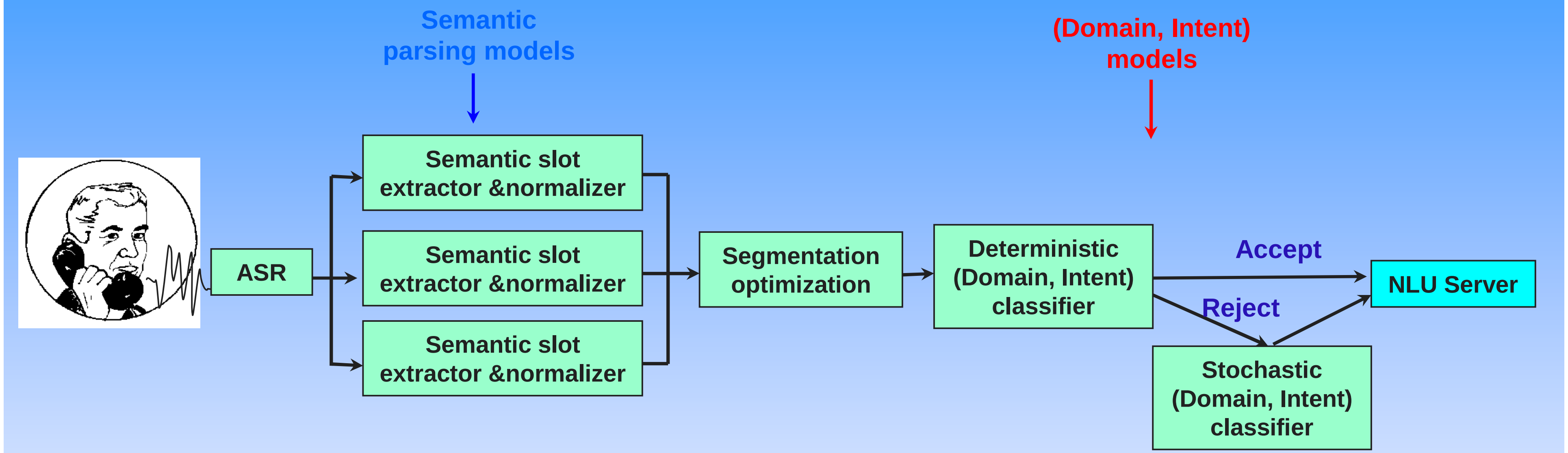
- Query templates ordered decreasingly by their coverage
- Intent/domain annotation of the most frequent templates
  - <RESTAURANT> <NEAR> <LOCATION> → find_FoodEstablisment
- Each annotated query template automatically translated into a parsing rule
- All queries covered by the annotated templates can be used as training for a statistical intent/domain classifier
- Desired behavior not covered by data is implemented as handwritten parsing rules

Parsing-based one-shot NLU architecture


Semantic parsing models
(Domain, Intent) model
ASR
Semantic slot extractor &normalizer
(Domain, Intent) classifier
NLU Server

- Bottom-up data processing
- Parsing models are context-dependent grammars based on semantic/name entity dictionaries
- Joint (Domain, Intent) classifier using semantic entity features
- Uses name entity dictionaries rather than manually annotated queries

Using multiple slot extractors and classifiers


Semantic parsing models
(Domain, Intent) models
ASR
Semantic slot extractor &normalizer
Semantic slot extractor &normalizer
Semantic slot extractor &normalizer
Segmentation optimization
Deterministic (Domain, Intent) classifier
Accept
Reject
NLU Server
Stochastic (Domain, Intent) classifier

- Slot extractors can be different parsers or even stochastic sequential annotators (e.g. current CRF,IBM Sire)
- Segmentation optimization: minimize (#semantic entities, #non-covered words)
- Deterministic classifier: Hash table or Nearest Neighbor
- Deterministic processing path needed for fixing the errors

Comparison with CRF annotation on the Calling domain Methodology

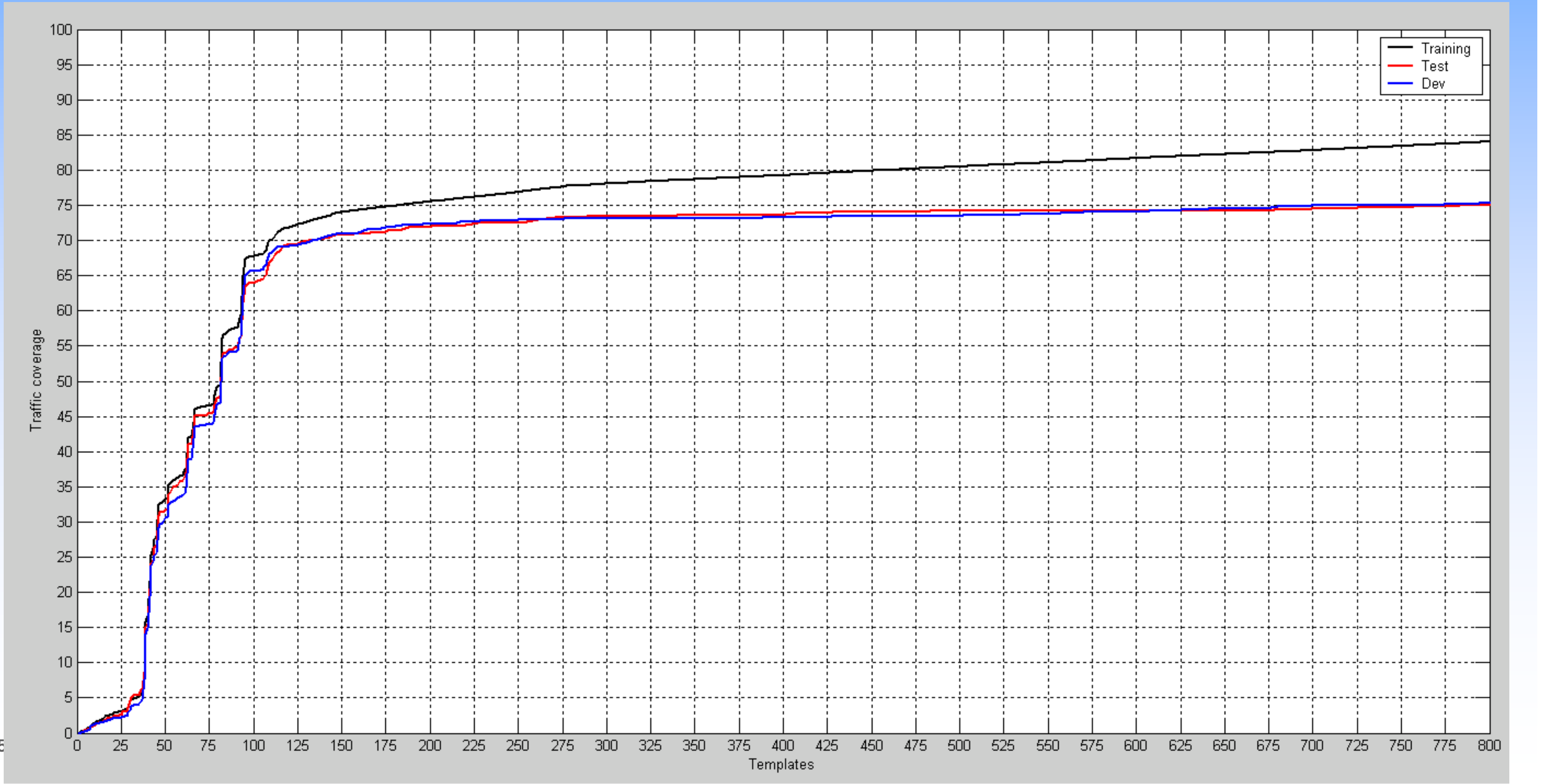
- Training set: 9248 sentences
- Concept discovery + grammar building = 20h (covers 2/3 traffic)
  - 60 grammar rules
  - Perl script top-down parser (handles free text) = 100 lines
  - Simple rule classifier [Parse => Intent] = 40 lines
- Mapping slots/intents into CRF’s annotation schema = 25h
- Template calibration (removing consistent annotation differences) = 20h

  <destination_phone>home</destination_phone> phone

  <destination_phone>home phone</destination_phone>
- Sort training templates by coverage/’confidence’ and downgrade the ones with inconsistent manual annotation
- Compute results on the dev/test sets for increasing coverage levels

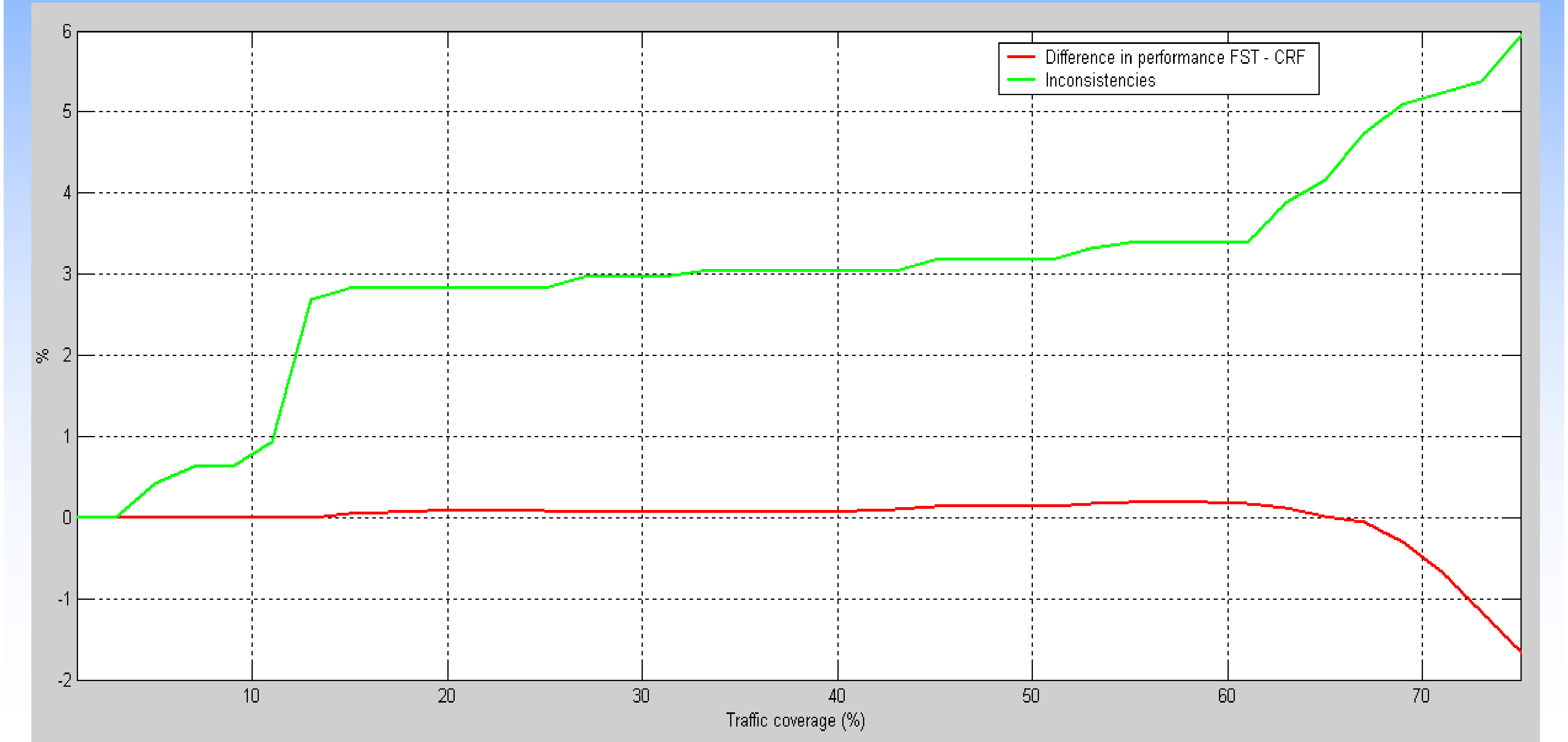
Coverage of the Dev/Test sets by the training templates

- Top 100 training templates cover roughly 2/3 of the traffic
- Fixing the training templates decreases coverage by <5% abs on the head of the distribution
  - Acc (TrainCov = X) >= Acc(TestCov = X%) >= Acc (TrainCov = X+5%)

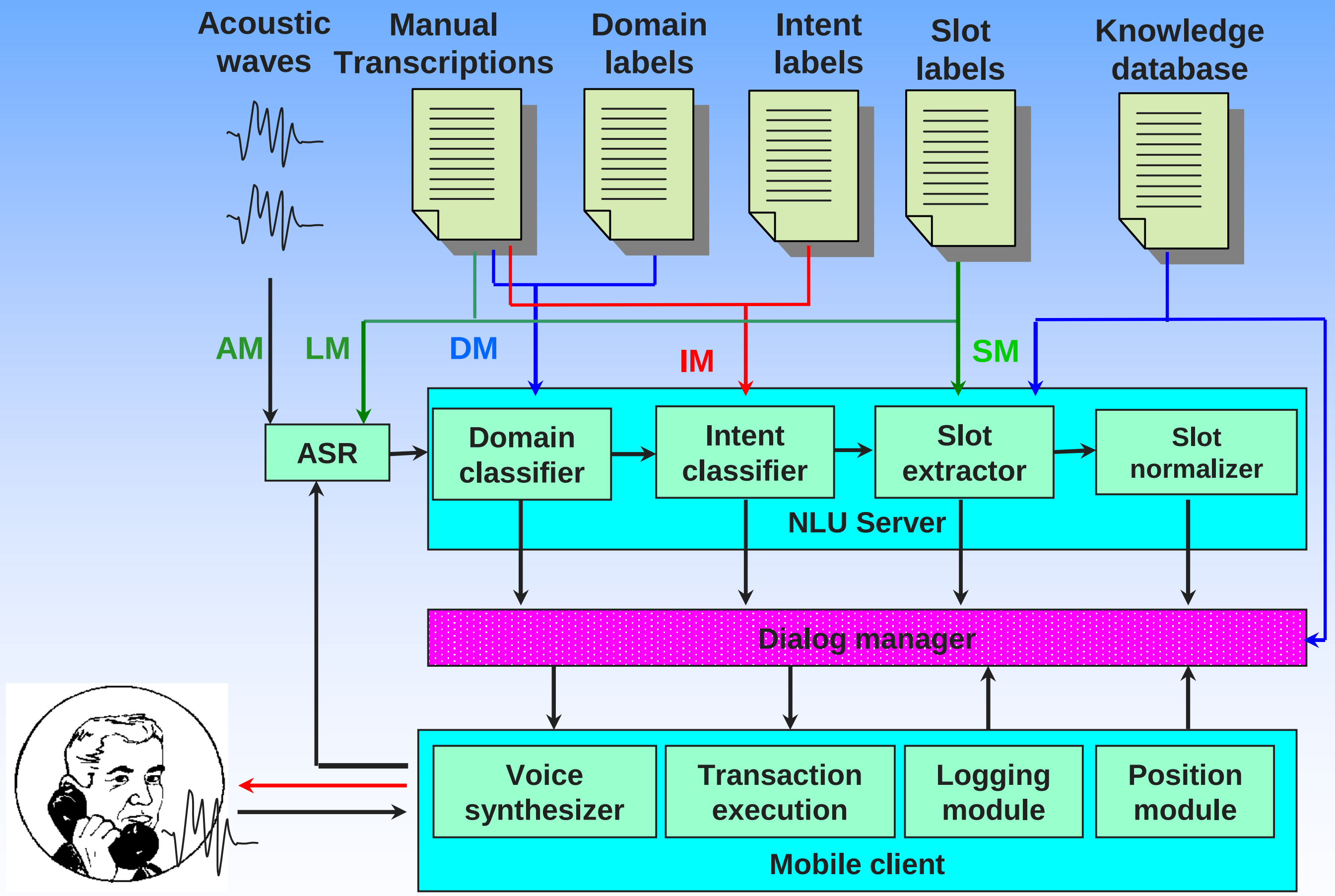


Difference in performance Grammar - CRF

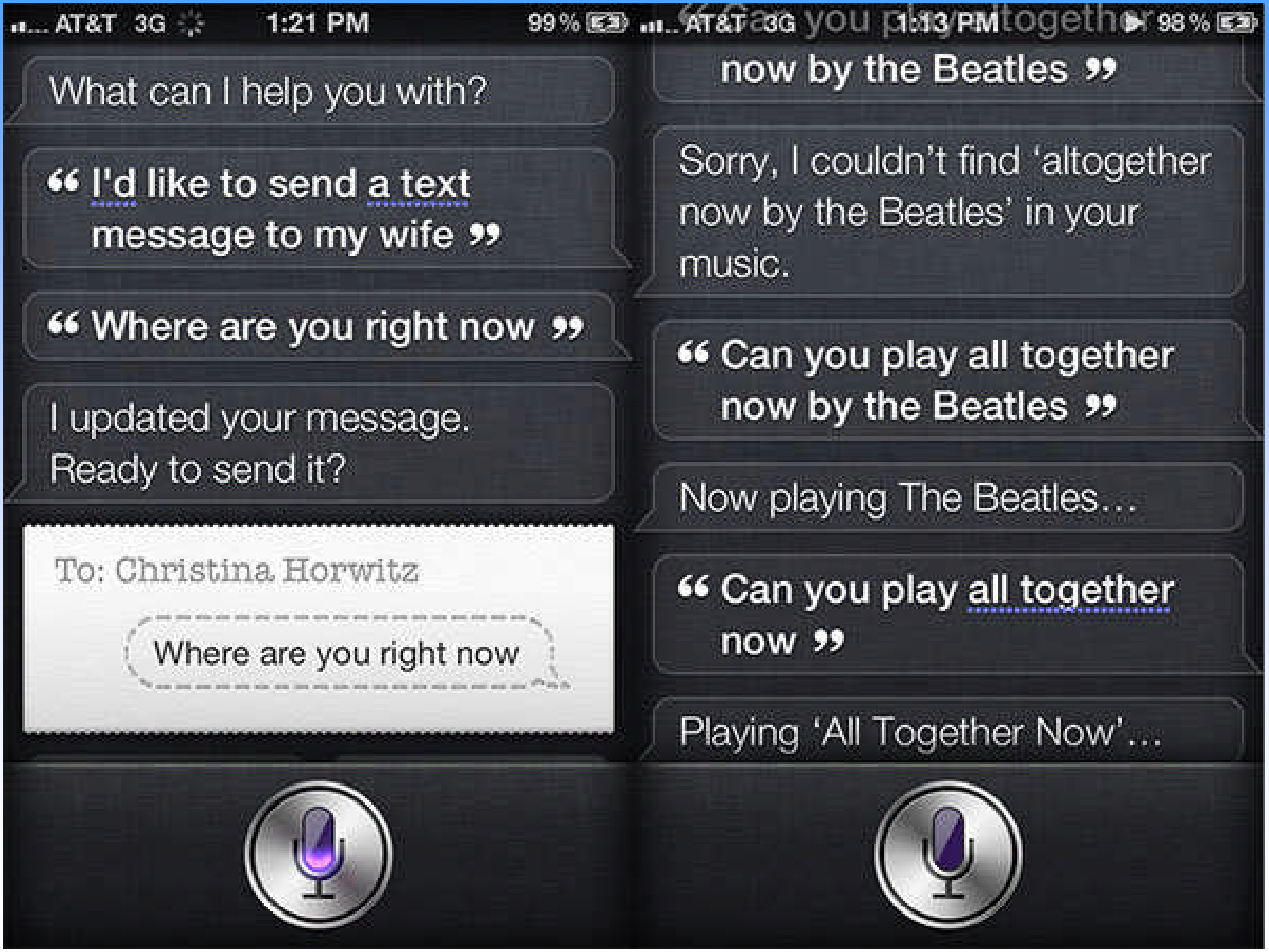
- On the top 15% of the traffic both methods give identical results
- On the next 50% of the traffic grammar slightly more accurate
- On the 1/3 traffic tail CRF is more accurate
  - A lot more annotation inconsistencies on complex templates



Client server NLU + dialog architecture



Client server NLU + dialog



What can I help you with?

“I'd like to send a text message to my wife”

“Where are you right now”

I updated your message. Ready to send it?

To: Christina Horwitz

Where are you right now

“Can you play altogether now by the Beatles”

Sorry, I couldn't find ‘altogether now by the Beatles’ in your music.

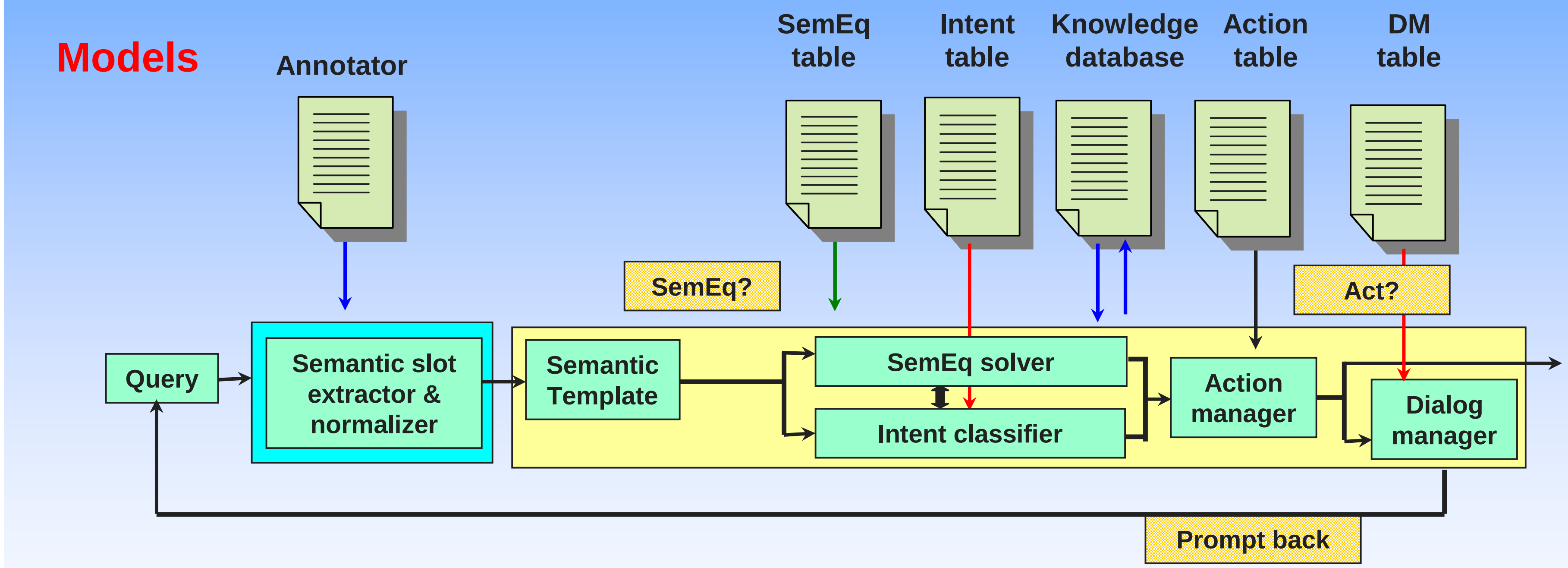
“Can you play all together now by the Beatles”

Now playing The Beatles…

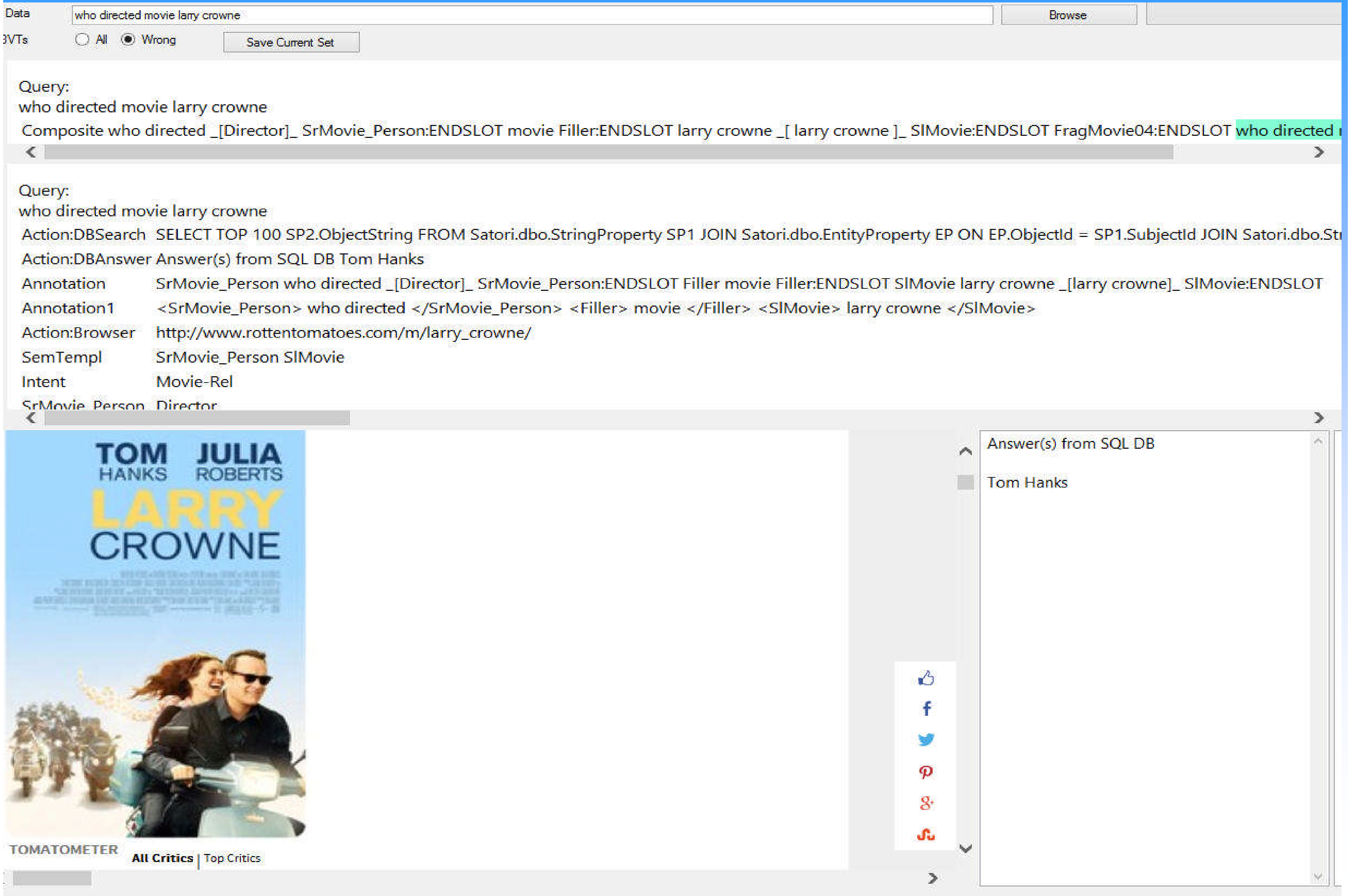
“Can you play all together now”

Playing ‘All Together Now’…

NLU + Question Answering + Dialog architecture



NLU + Question Answering + Dialog



Data: who directed movie larry crowne

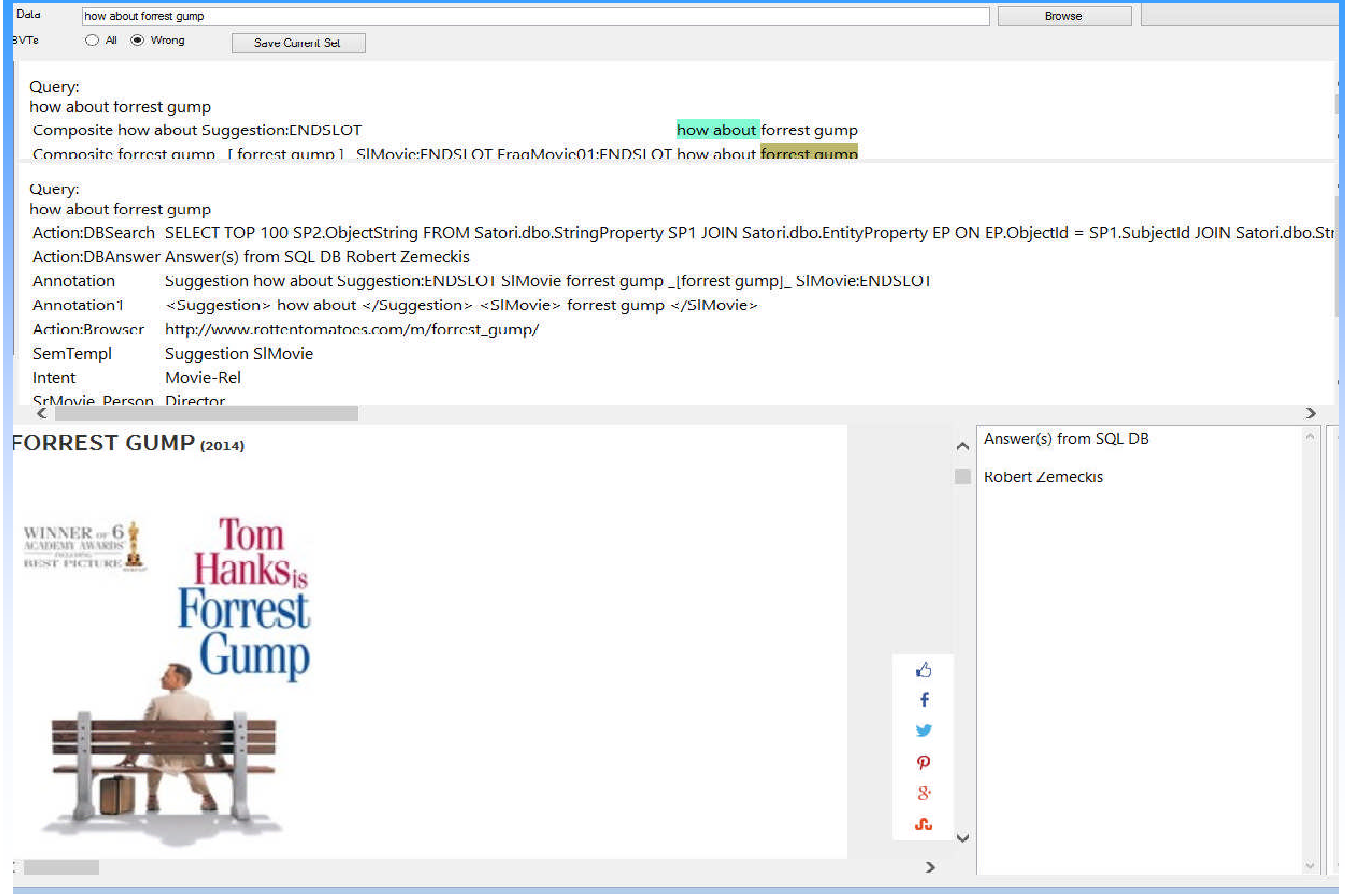
```
Query:
who directed movie larry crowne
 Composite who directed _[Director]_ SrMovie_Person:ENDSLOT movie Filler:ENDSLOT larry crowne _[ larry crowne ]_ SlMovie:ENDSLOT FragMovie04:ENDSLOT who directed 
```

```
Query:
who directed movie larry crowne
 Action:DBSearch  SELECT TOP 100 SP2.ObjectString FROM Satori.dbo.StringProperty SP1 JOIN Satori.dbo.EntityProperty EP ON EP.ObjectId = SP1.SubjectId JOIN Satori.dbo.St
 Action:DBAnswer Answer(s) from SQL DB Tom Hanks
 Annotation       SrMovie_Person who directed _[Director]_ SrMovie_Person:ENDSLOT Filler movie Filler:ENDSLOT SlMovie larry crowne _[larry crowne]_ SlMovie:ENDSLOT
 Annotation1      <SrMovie_Person> who directed </SrMovie_Person> <Filler> movie </Filler> <SlMovie> larry crowne </SlMovie>
 Action:Browser   http://www.rottentomatoes.com/m/larry_crowne/
 SemTempl         SrMovie_Person SlMovie
 Intent           Movie-Rel
 SrMovie_Person  Director
```

Answer(s) from SQL DB

Tom Hanks

NLU + Question Answering + Dialog



```
Query:
how about forrest gump
Composite how about Suggestion:ENDSLOT                                    how about forrest gump
Composite forrest gump  [ forrest gump ]  SlMovie:ENDSLOT FragMovie01:ENDSLOT how about forrest gump
```

```
Query:
how about forrest gump
Action:DBSearch  SELECT TOP 100 SP2.ObjectString FROM Satori.dbo.StringProperty SP1 JOIN Satori.dbo.EntityProperty EP ON EP.ObjectId = SP1.SubjectId JOIN Satori.dbo.Str
Action:DBAnswer Answer(s) from SQL DB Robert Zemeckis
Annotation       Suggestion how about Suggestion:ENDSLOT SlMovie forrest gump _[forrest gump]_ SlMovie:ENDSLOT
Annotation1      <Suggestion> how about </Suggestion> <SlMovie> forrest gump </SlMovie>
Action:Browser   http://www.rottentomatoes.com/m/forrest_gump/
SemTempl         Suggestion SlMovie
Intent           Movie-Rel
SrMovie_Person   Director
```

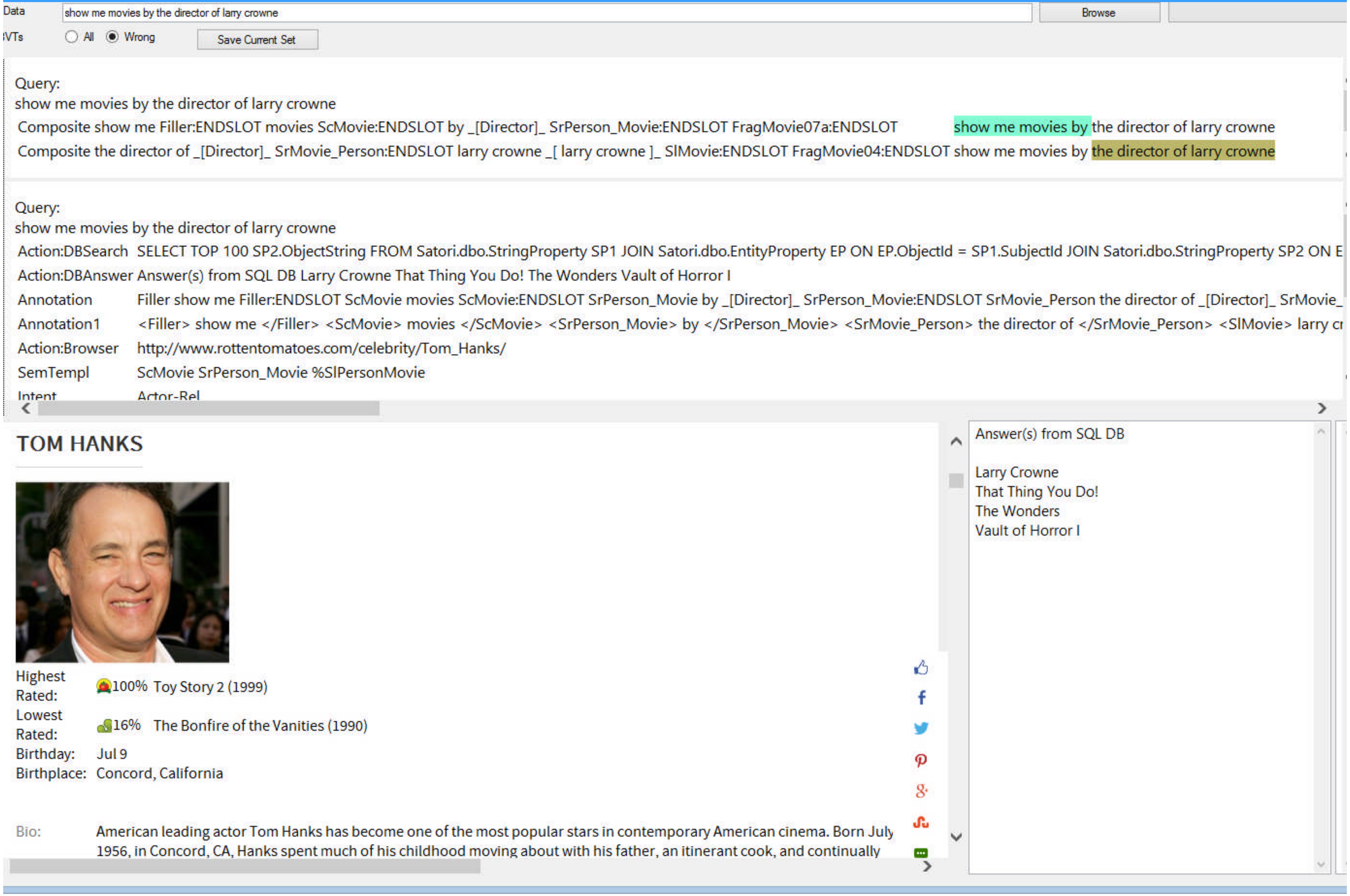
FORREST GUMP (2014)

WINNER OF 6 ACADEMY AWARDS INCLUDING BEST PICTURE
Tom Hanks is Forrest Gump

Answer(s) from SQL DB

Robert Zemeckis

NLU + Question Answering + Dialog



Query:
show me movies by the director of larry crowne
Composite show me Filler:ENDSLOT movies ScMovie:ENDSLOT by _[Director]_ SrPerson_Movie:ENDSLOT FragMovie07a:ENDSLOT show me movies by the director of larry crowne
Composite the director of _[Director]_ SrMovie_Person:ENDSLOT larry crowne _[larry crowne]_ SlMovie:ENDSLOT FragMovie04:ENDSLOT show me movies by the director of larry crowne

Query:
show me movies by the director of larry crowne
Action:DBSearch SELECT TOP 100 SP2.ObjectString FROM Satori.dbo.StringProperty SP1 JOIN Satori.dbo.EntityProperty EP ON EP.ObjectId = SP1.SubjectId JOIN Satori.dbo.StringProperty SP2 ON E
Action:DBAnswer Answer(s) from SQL DB Larry Crowne That Thing You Do! The Wonders Vault of Horror I
Annotation Filler show me Filler:ENDSLOT ScMovie movies ScMovie:ENDSLOT SrPerson_Movie by _[Director]_ SrPerson_Movie:ENDSLOT SrMovie_Person the director of _[Director]_ SrMovie_
Annotation1 <Filler> show me </Filler> <ScMovie> movies </ScMovie> <SrPerson_Movie> by </SrPerson_Movie> <SrMovie_Person> the director of </SrMovie_Person> <SlMovie> larry cr
Action:Browser http://www.rottentomatoes.com/celebrity/Tom_Hanks/
SemTempl ScMovie SrPerson_Movie %SlPersonMovie
Intent Actor-Rel

TOM HANKS

Highest Rated: 100% Toy Story 2 (1999)
Lowest Rated: 16% The Bonfire of the Vanities (1990)
Birthday: Jul 9
Birthplace: Concord, California

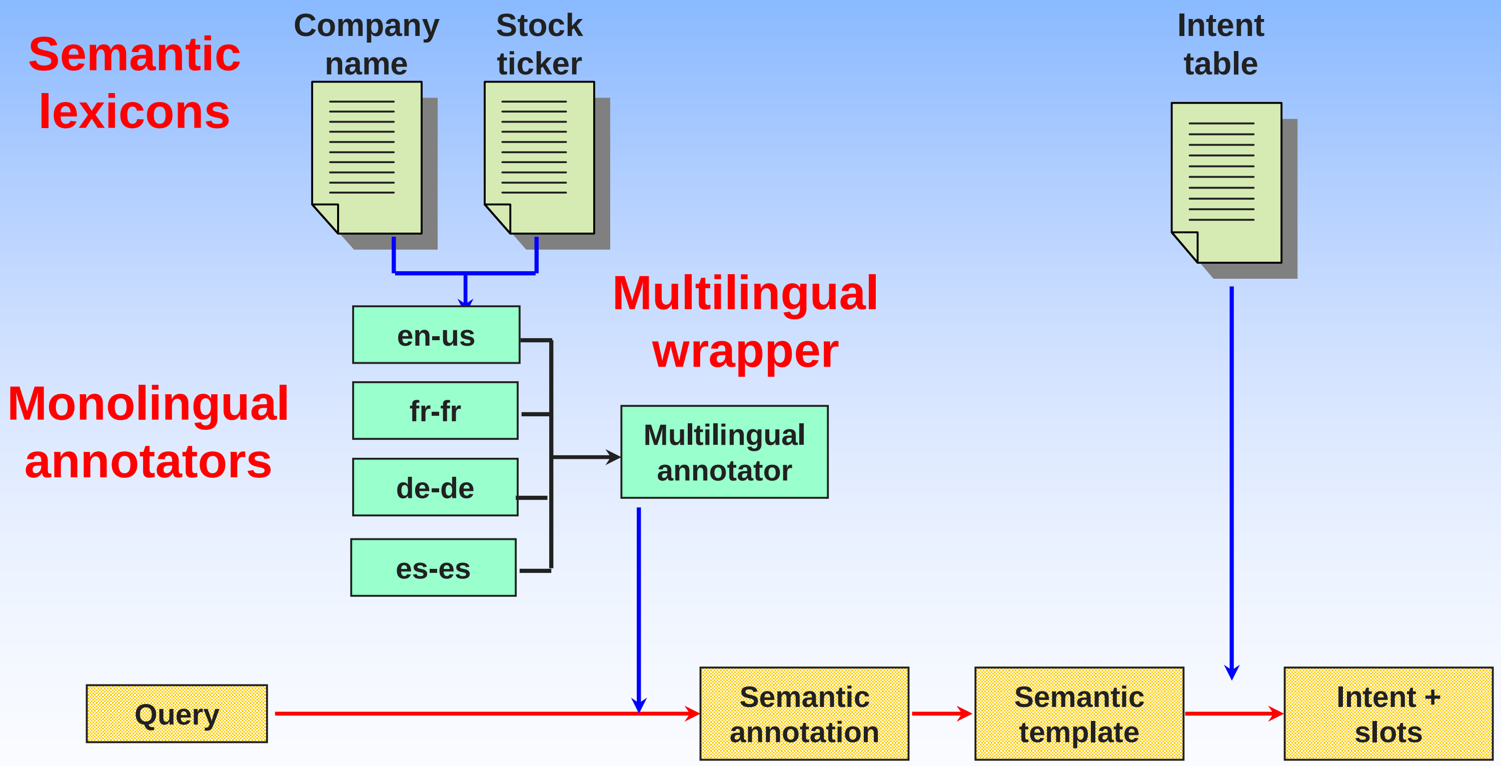
Bio: American leading actor Tom Hanks has become one of the most popular stars in contemporary American cinema. Born July 1956, in Concord, CA, Hanks spent much of his childhood moving about with his father, an itinerant cook, and continually

Answer(s) from SQL DB

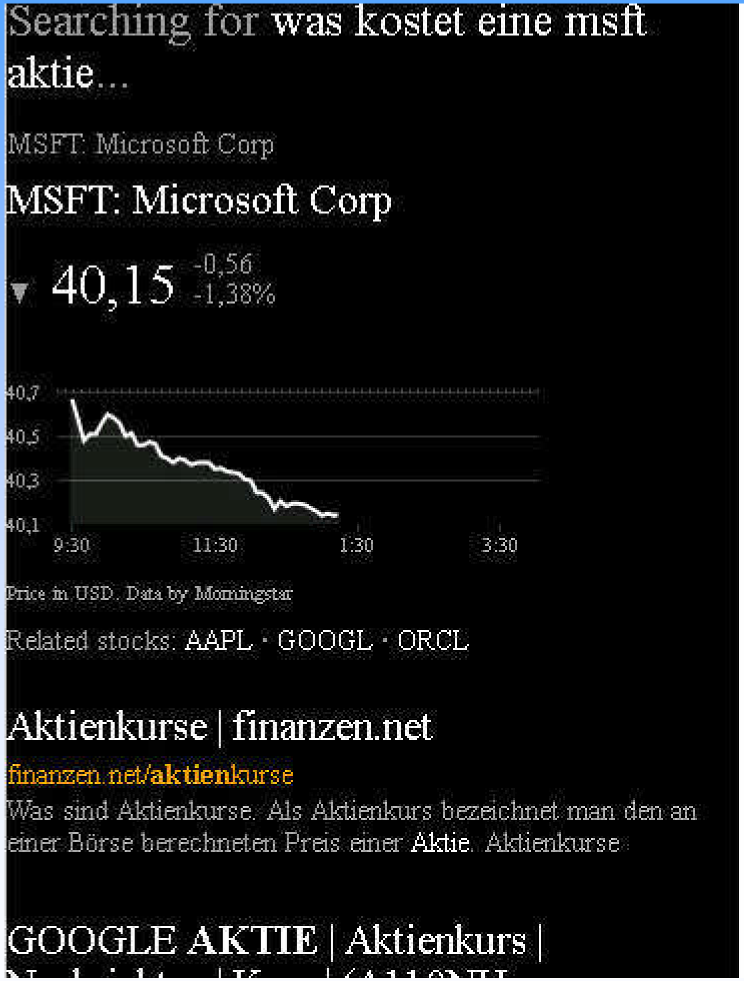
Larry Crowne
That Thing You Do!
The Wonders
Vault of Horror I

Multilingual query understanding architecture

- **Monolingual annotators can be combined into a Multilingual annotator**



Multilingual query understanding



Searching for was kostet eine msft aktie...

MSFT: Microsoft Corp

MSFT: Microsoft Corp

▼ 40,15 -0,56 -1,38%

Price in USD. Data by Morningstar

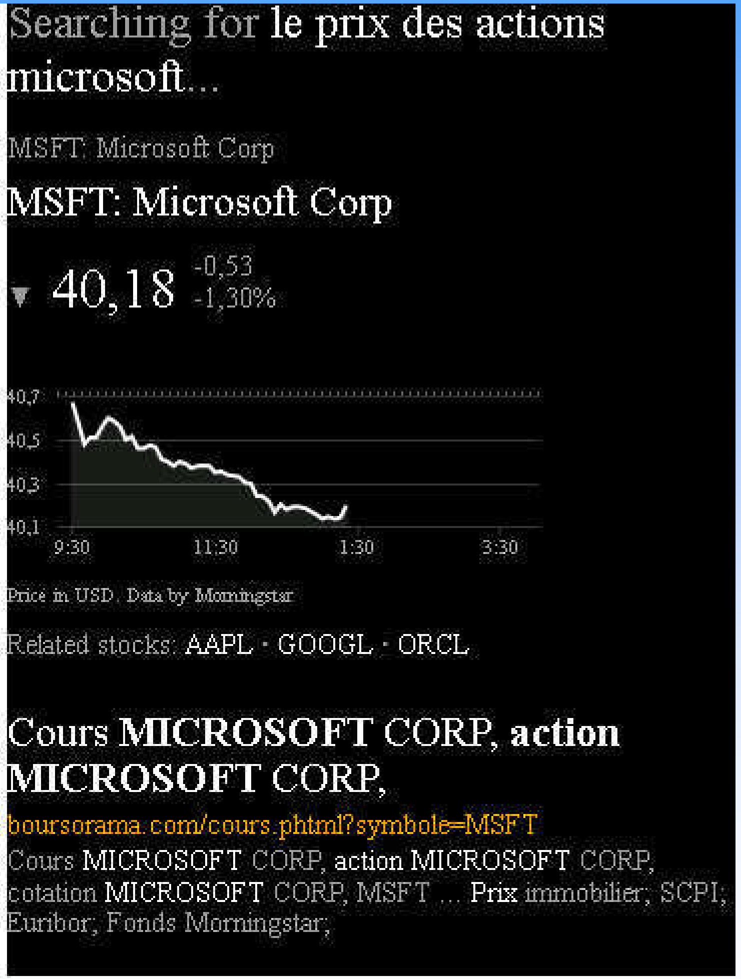
Related stocks: AAPL · GOOGL · ORCL

Aktienkurse | finanzen.net

finanzen.net/**aktien**kurse

Was sind Aktienkurse. Als Aktienkurs bezeichnet man den an einer Börse berechneten Preis einer Aktie. Aktienkurse

GOOGLE AKTIE | Aktienkurs |



Searching for le prix des actions microsoft...

MSFT: Microsoft Corp

MSFT: Microsoft Corp

▼ 40,18 -0,53 -1,30%

Price in USD. Data by Morningstar

Related stocks: AAPL · GOOGL · ORCL

Cours **MICROSOFT** CORP, **action MICROSOFT** CORP,

boursorama.com/cours.phtml?symbole=MSFT

Cours MICROSOFT CORP, action MICROSOFT CORP, cotation MICROSOFT CORP, MSFT ... Prix immobilier, SCPI, Euribor, Fonds Morningstar,

Bottom-up vs. top down parsing

- Bottom-up parsing computes the semantic structure from the query
- Top-down parsing checks whether the query is covered by an imposed semantic structure (including guessed entities)

- Bottom-up parsing

  watch family history season nine on netflix

- Top-down parsing

Handwritten regexp “Stream *{1,3} ShowContext URL*”

watch family history season nine on netflix

- Preferred top-down parse since it only misses two words

Legend: Stream ShowTitle Filler ShowContext URL

Context-based semantic disambiguation

- Some entities can be labeled along with contextual phrases
  - "red sox news" vs. "obama news"
  - Contextual phrases can be labeled un-ambiguosly
  - Increase both coverage and accuracy
- Sometimes both the main entity and the contextual phrase are ambiguous
  - "Alice in wonderland": Book title, Movie, Song, Album
  - "3-D" is also ambiguous
  - "Alice in wonderland 3-D": Not ambiguous anymore
- Ambiguous entities need to be disambiguated by contextual phrases
- Many ambiguous entity names in content databases

Increasing parser coverage

- Increasing the query-to-template compression rate
  - Making matching algorithms insensitive to “-“ vs “ “ vs “” and “’s” vs “s” vs “” (“king’s speech”,” kings speech”,” king speech” considered the same)
  - Modeling contextual phrases: “avatar movie in 3-d”
- Increasing the number of labeled query templates
  - Automatic labeling is possible to some extent since some entities and/or entity ordering do not contribute to the intent assignment decision
  - <BUSINESS> <NEAR>
    <Find> <BUSINESS> <NEAR> → ***find_LocalBusiness***
    <Find> <NEAR> < BUSINESS >
  - The number of templates manually labeled is much smaller if using only reduced templates
- Increasing the number of semantic entities modeled

Issues with automated entity updates

1. Name confusion
   - “Moby Dick”
   - It is hard to compute confusability based solely on the field data.
2. Differences between the listing names and the actual name requests
   - “On the border mexican grill & cantina” => “on the border”, “on the border restaurant”
   - “Sears Roebuck and Co” => “sears”, “sears store”
3. Differences between what is spoken and what is recognized
   - “fry’s electronics” => “fries electronics” “toysRus” => “toys are us”

Issues with automated concept name updates

4. Dependence on the final website/search engine
   - Yelp chokes on “cvs pharmacy”/”bertucci’s pizzeria”; expects just “cvs”/”bertucci’s”
   - IMDB needs the exact movie ID as in its database in order to go directly to the page

- For regular automated updates items 2-4 are hard to anticipate

Advantages of including a parsing-based deterministic component

- Trained with little manually annotated data
  - No manual transcription or semantic labeling is necessary
  - The classifier training set is bootstrapped from the fully abstracted queries
- Flexible to:
  - Adding coverage for semantic entities not seen in the data
  - Name guessing
  - Changing the granularity of the semantic interpretation
  - Closely controlling system behavior
- Disadvantages
  - Handwritten rules more difficult to maintain

User behavior analysis

- Most people repeat the query than correct the recognition output
  - Some users eventually get correct recognition after a few trials
  - Some users try many times and don't get it right due to OOVs
    - If not correct after the 5th trial, the likelihood of eventually getting it right < 5%
    - Including repeated queries highly biases system stats (e.g. OOV)

- Users would rather type queries longer than 7 words: "Say What? Why users choose to speak their web queries", M. Kamvar, D. Beeferman (Google)
  - Difficult to fluently voice a large amount of info in single query

- User gaming / testing
  - Non-native speakers passing the phone to other (native) speakers
  - Repeated queries on misrecognitions but correct document retrieved

“Say What? Why users choose to speak their web queries”, M. Kamvar, D. Beeferman (Google)

- Analyze the factors that are correlated with a decision to speak a web search query rather than type it.
  - Experiments using Google Mobile Application on Blackberry
  - 75K users, 1M+ queries using both typing & voice search
- Keyboard type: P(V|FK) = .346   P(V|CK) = .416
- Query length: more likely to speak a query shorter than 6 words than a longer query
  - Possibly determined by the extent to which users need to remember speech queries in an “articulatory buffer” prior to speaking
  - In our data: Longer queries are 10-20x less frequent than 1-3 word queries and many not really used for search
- Query popularity/frequency (using completion suggestion feature): Not correlated

- Query semantics: DA queries more likely to be spoken
  - Heavily uses “local/near” search (using GPS location)
- Spoken queries trigger “quick results” (no need to click) 12% more often than typed queries
  - Users speak their queries in situations where the entire search experience will be “hands-free”
  - Half of Maps queries are spoken
- Factor proposed but not analyzed:
  - Users’ situational context: their primary activity at the time of querying
  - More likely to use voice when driving/walking than when riding subway/bus or in a meeting
  - Logging user’s velocity

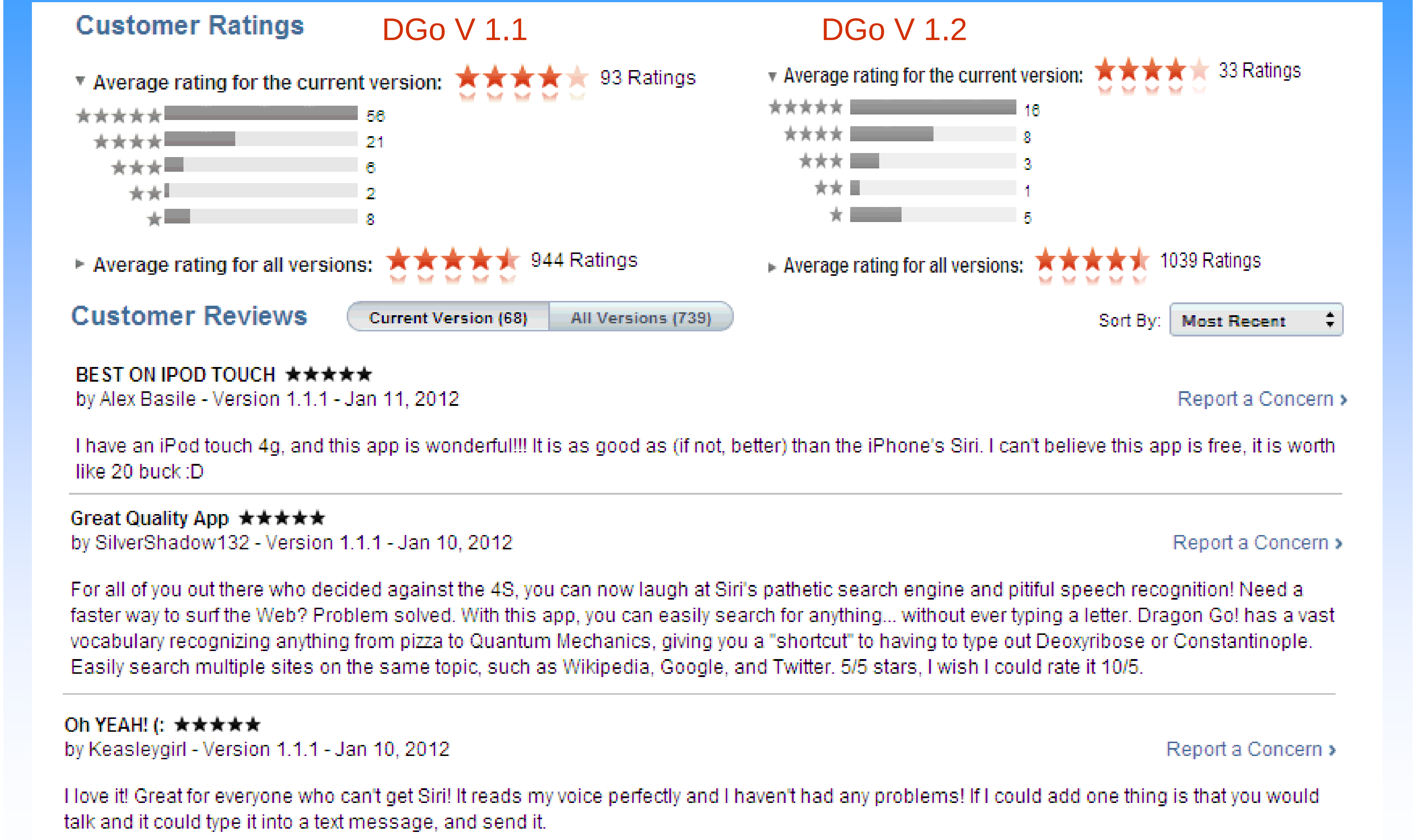
How errors are perceived by humans

- If the feature space of a ML system is "humanly understandable" then some errors may look very embarrassing and there will be high pressure to guarantee they won't happen again
  - ➢ "The most humiliating moment in my writing career was referring to Warren Buffett and Peter Lynch as 'Buffet and Lunch' not in a column, but in my book, Your Next Great Stock. That's because I was too busy and greedy to take a book sabbatical and instead wrote the thing at night using Dragon software" ("New iPhone Bodes Well for Speech Stock", Smart Money magazine)

VS.

"I got another chuckle when I asked to 'Search for the square root of 155' and it asked me which address '155 Root Ave' was the one I wanted" (quote from "Apple's Siri Versus Dragon Go! and Vlingo" review in PC magazine)

User ratings for voice search technologies



Customer Ratings

DGo V 1.1

▾ Average rating for the current version: ★★★★☆ 93 Ratings

★★★★★ 56
★★★★ 21
★★★ 6
★★ 2
★ 8

▸ Average rating for all versions: ★★★★½ 944 Ratings

DGo V 1.2

▾ Average rating for the current version: ★★★★☆ 33 Ratings

★★★★★ 16
★★★★ 8
★★★ 3
★★ 1
★ 5

▸ Average rating for all versions: ★★★★½ 1039 Ratings

Customer Reviews

BEST ON IPOD TOUCH ★★★★★
by Alex Basile - Version 1.1.1 - Jan 11, 2012

Report a Concern ›

I have an iPod touch 4g, and this app is wonderful!!! It is as good as (if not, better) than the iPhone's Siri. I can't believe this app is free, it is worth like 20 buck :D

Great Quality App ★★★★★
by SilverShadow132 - Version 1.1.1 - Jan 10, 2012

Report a Concern ›

For all of you out there who decided against the 4S, you can now laugh at Siri's pathetic search engine and pitiful speech recognition! Need a faster way to surf the Web? Problem solved. With this app, you can easily search for anything... without ever typing a letter. Dragon Go! has a vast vocabulary recognizing anything from pizza to Quantum Mechanics, giving you a "shortcut" to having to type out Deoxyribose or Constantinople. Easily search multiple sites on the same topic, such as Wikipedia, Google, and Twitter. 5/5 stars, I wish I could rate it 10/5.

Oh YEAH! (: ★★★★★
by Keasleygirl - Version 1.1.1 - Jan 10, 2012

Report a Concern ›

I love it! Great for everyone who can't get Siri! It reads my voice perfectly and I haven't had any problems! If I could add one thing is that you would talk and it could type it into a text message, and send it.

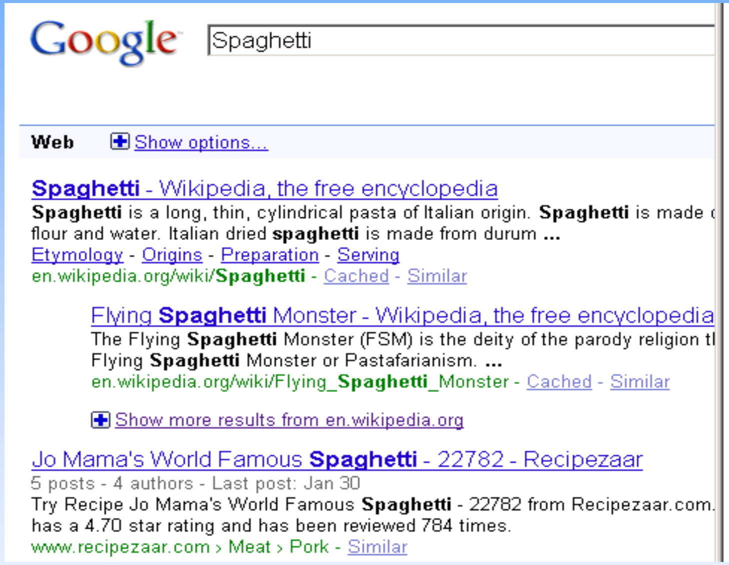
Building large scale voice-search systems: caveats

- Being able to fully fix the semantic specs of the app before building it is a myth
- Manual labeling of a large amount of individual queries with high consistency/accuracy, reasonable semantic granularity in a reasonable amount of time/costs is a myth
- If buggy, the client app may be a strong reason for user annoyance. Very expensive to recall/update.
- Need to keep full logs of all ASR/NLU intermediate results

Semantic web-search

- T. Imielinski & A. Signorini: “If you ask nicely, I will answer: Semantic Search and Today’s Search Engines”, 3rd IEEE International Conference on Semantic Computing (2009)

- Search engines sensitive to the way queries are constructed

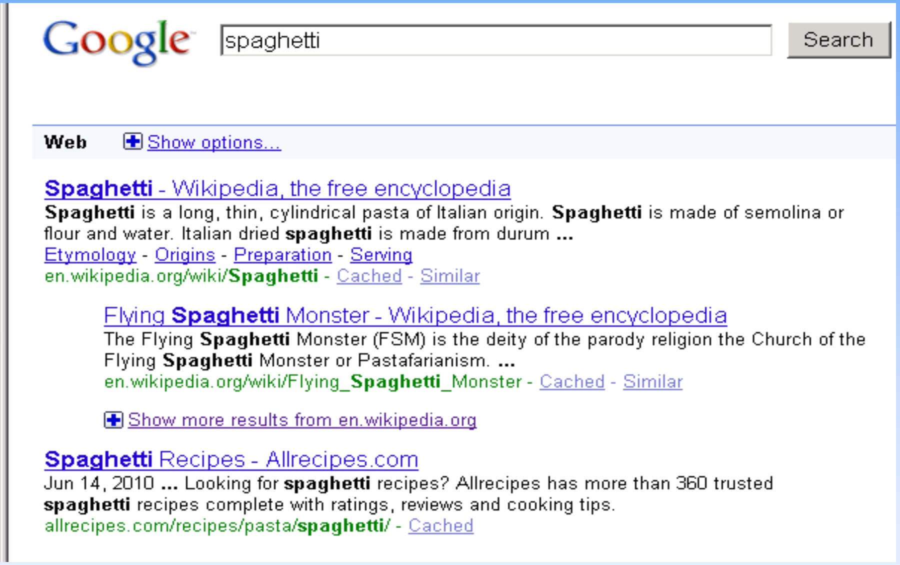


Spaghetti - Wikipedia, the free encyclopedia
Spaghetti is a long, thin, cylindrical pasta of Italian origin. **Spaghetti** is made o
flour and water. Italian dried **spaghetti** is made from durum **...**
Etymology - Origins - Preparation - Serving
en.wikipedia.org/wiki/**Spaghetti** - Cached - Similar

Flying **Spaghetti** Monster - Wikipedia, the free encyclopedia
The Flying **Spaghetti** Monster (FSM) is the deity of the parody religion tl
Flying **Spaghetti** Monster or Pastafarianism. **...**
en.wikipedia.org/wiki/Flying_**Spaghetti**_Monster - Cached - Similar

Show more results from en.wikipedia.org

Jo Mama's World Famous **Spaghetti** - 22782 - Recipezaar
5 posts - 4 authors - Last post: Jan 30
Try Recipe Jo Mama's World Famous **Spaghetti** - 22782 from Recipezaar.com.
has a 4.70 star rating and has been reviewed 784 times.
www.recipezaar.com › Meat › Pork - Similar



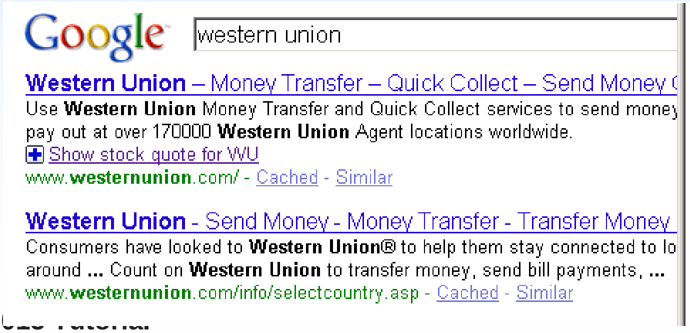
Spaghetti - Wikipedia, the free encyclopedia
Spaghetti is a long, thin, cylindrical pasta of Italian origin. **Spaghetti** is made of semolina or
flour and water. Italian dried **spaghetti** is made from durum **...**
Etymology - Origins - Preparation - Serving
en.wikipedia.org/wiki/**Spaghetti** - Cached - Similar

Flying **Spaghetti** Monster - Wikipedia, the free encyclopedia
The Flying **Spaghetti** Monster (FSM) is the deity of the parody religion the Church of the
Flying **Spaghetti** Monster or Pastafarianism. **...**
en.wikipedia.org/wiki/Flying_**Spaghetti**_Monster - Cached - Similar

Show more results from en.wikipedia.org

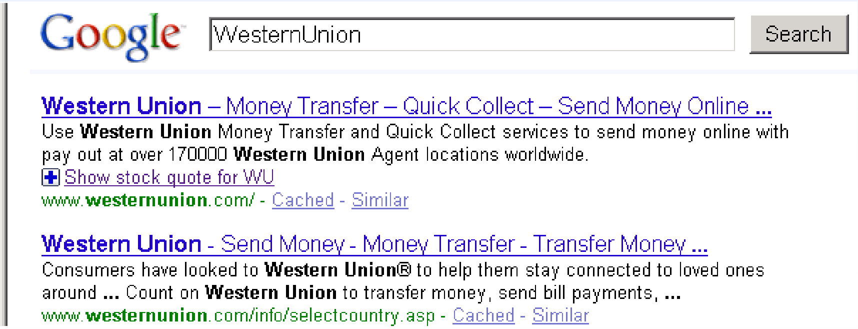
Spaghetti Recipes - Allrecipes.com
Jun 14, 2010 **...** Looking for **spaghetti** recipes? Allrecipes has more than 360 trusted
spaghetti recipes complete with ratings, reviews and cooking tips.
allrecipes.com/recipes/pasta/**spaghetti**/ - Cached

- Popular queries with only one right answer are well served



Western Union – Money Transfer – Quick Collect – Send Money O
Use **Western Union** Money Transfer and Quick Collect services to send money
pay out at over 170000 **Western Union** Agent locations worldwide.
Show stock quote for WU
www.**westernunion**.com/ - Cached - Similar

Western Union - Send Money - Money Transfer - Transfer Money
Consumers have looked to **Western Union**® to help them stay connected to lo
around **...** Count on **Western Union** to transfer money, send bill payments, **...**
www.**westernunion**.com/info/selectcountry.asp - Cached - Similar



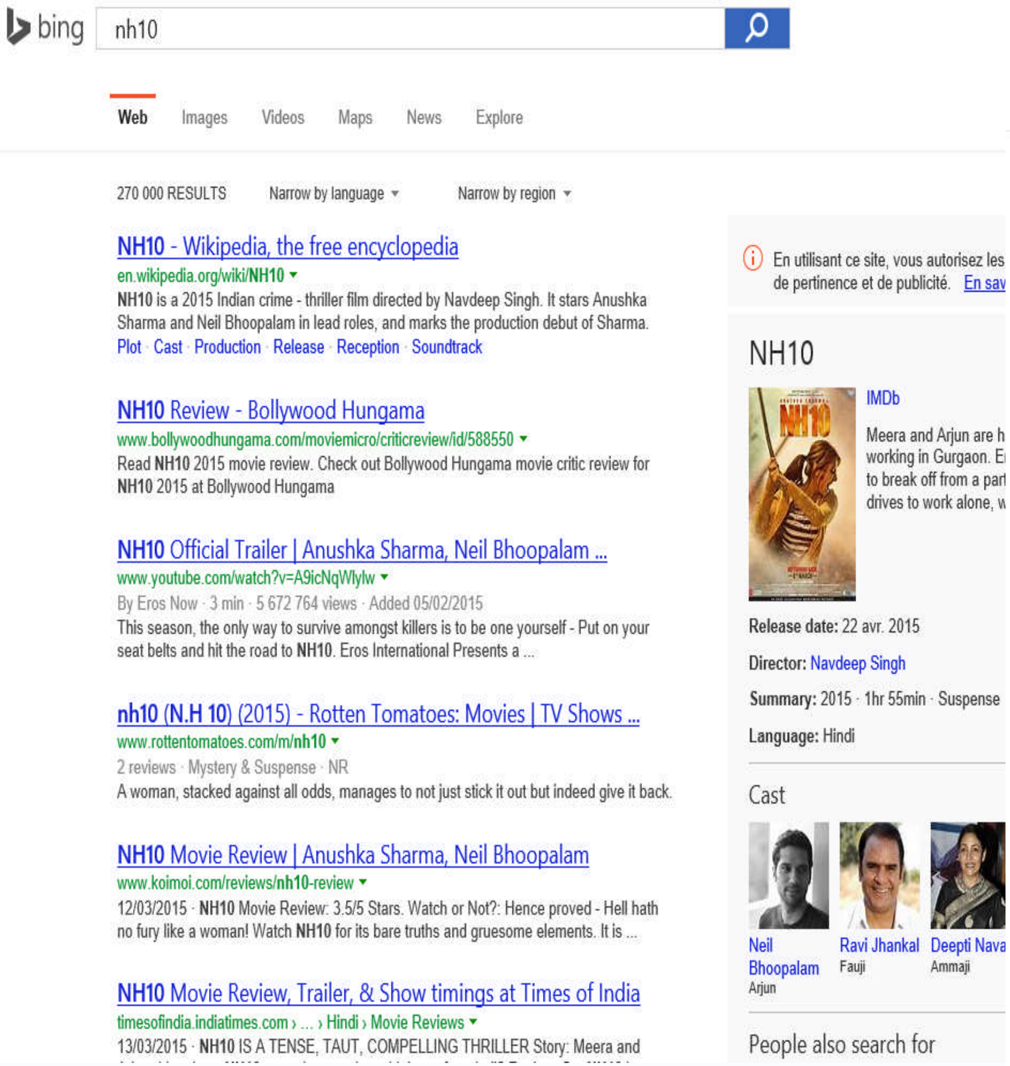
Western Union – Money Transfer – Quick Collect – Send Money Online **...**
Use **Western Union** Money Transfer and Quick Collect services to send money online with
pay out at over 170000 **Western Union** Agent locations worldwide.
Show stock quote for WU
www.**westernunion**.com/ - Cached - Similar

Western Union - Send Money - Money Transfer - Transfer Money **...**
Consumers have looked to **Western Union**® to help them stay connected to loved ones
around **...** Count on **Western Union** to transfer money, send bill payments, **...**
www.**westernunion**.com/info/selectcountry.asp - Cached - Similar

Semantic web-search

- Search engines remain many times keyword oriented
  - Helped by Internet’s redundancy of information and user generated content
  - The burden of selecting the right keywords is left to the user
- Distinction between understanding a query and being able to answer it
- Semantic engine: invariant to the way the query is formulated (rephrase)
  - Many academic/industrial initiatives to make the web semantic (W3C Semantic Web Activity)
- Metrics to measure ”how semantic” a given search engine is
  - Entropy of Search Result Page
  - Top-K results overlap
  - One-Right-Answer Invariance: the fraction of queries for which the correct answer appears in the result page
- Query data: 40K automatically generated based on templates (“bio of person”)
  - Over-specifying the query (”France the country”): Top choice the same between 10-45% of the time, Top-5 choice 100% overlap almost never
  - Number transliteration (”top 20 cars”): Only 3% of the time top choice is the same
  - Rephrasing: 90% of the time the correct answer is eventually retrieved but Top-K results overlap is low

Semantic web-search



270 000 RESULTS Narrow by language ▾ Narrow by region ▾

NH10 - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/NH10 ▾
NH10 is a 2015 Indian crime - thriller film directed by Navdeep Singh. It stars Anushka Sharma and Neil Bhoopalam in lead roles, and marks the production debut of Sharma.
Plot · Cast · Production · Release · Reception · Soundtrack

NH10 Review - Bollywood Hungama
www.bollywoodhungama.com/moviemicro/criticreview/id/588550 ▾
Read NH10 2015 movie review. Check out Bollywood Hungama movie critic review for NH10 2015 at Bollywood Hungama

NH10 Official Trailer | Anushka Sharma, Neil Bhoopalam ...
www.youtube.com/watch?v=A9icNqWlylw ▾
By Eros Now · 3 min · 5 672 764 views · Added 05/02/2015
This season, the only way to survive amongst killers is to be one yourself - Put on your seat belts and hit the road to NH10. Eros International Presents a ...

nh10 (N.H 10) (2015) - Rotten Tomatoes: Movies | TV Shows ...
www.rottentomatoes.com/m/nh10 ▾
2 reviews · Mystery & Suspense · NR
A woman, stacked against all odds, manages to not just stick it out but indeed give it back.

NH10 Movie Review | Anushka Sharma, Neil Bhoopalam
www.koimoi.com/reviews/nh10-review ▾
12/03/2015 · NH10 Movie Review: 3.5/5 Stars. Watch or Not?: Hence proved - Hell hath no fury like a woman! Watch NH10 for its bare truths and gruesome elements. It is ...

NH10 Movie Review, Trailer, & Show timings at Times of India
timesofindia.indiatimes.com › ... › Hindi › Movie Reviews ▾
13/03/2015 · NH10 IS A TENSE, TAUT, COMPELLING THRILLER Story: Meera and

ⓘ En utilisant ce site, vous autorisez les de pertinence et de publicité. En sav

NH10

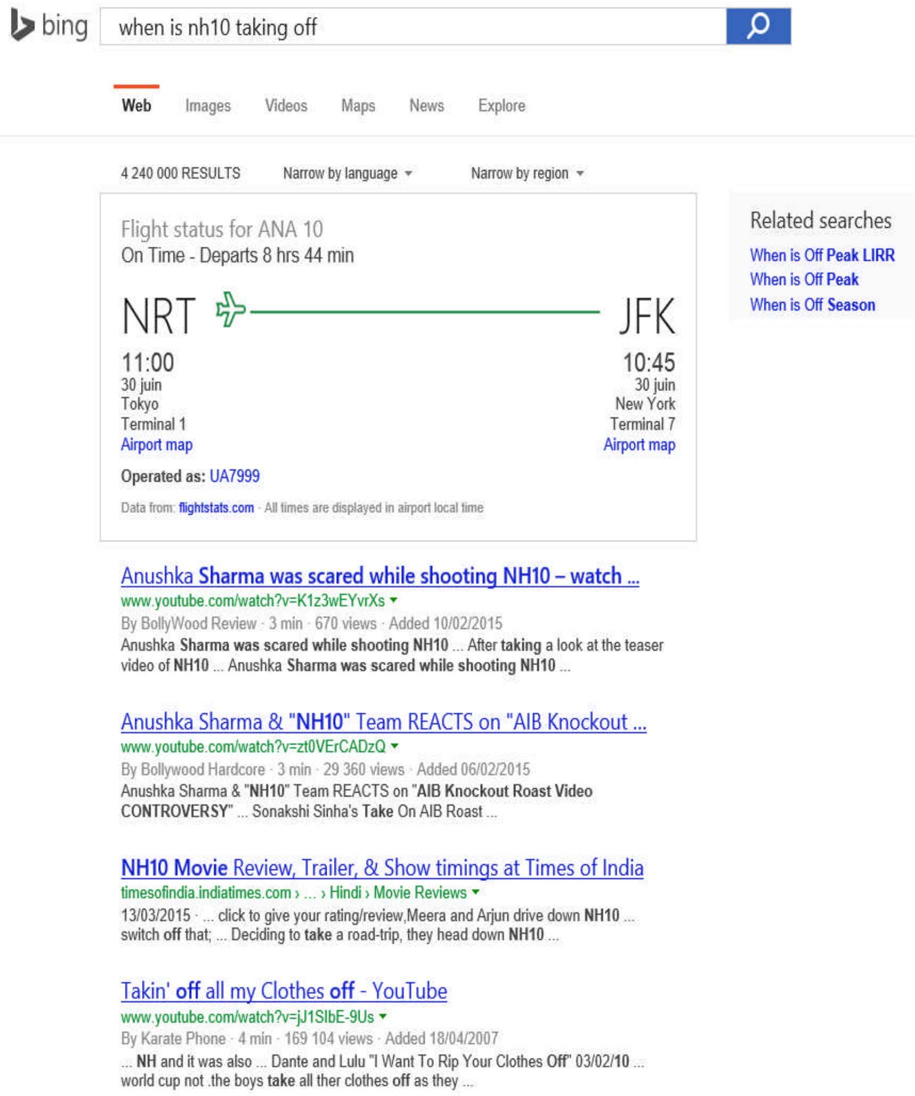
IMDb
Meera and Arjun are h working in Gurgaon. E to break off from a part drives to work alone, w

Release date: 22 avr. 2015
Director: Navdeep Singh
Summary: 2015 · 1hr 55min · Suspense
Language: Hindi

Cast

Neil Bhoopalam	Ravi Jhankal	Deepti Nava
Arjun	Fauji	Ammaji

People also search for



4 240 000 RESULTS Narrow by language ▾ Narrow by region ▾

Flight status for ANA 10
On Time - Departs 8 hrs 44 min

NRT	JFK
11:00	10:45
30 juin	30 juin
Tokyo	New York
Terminal 1	Terminal 7
Airport map	Airport map

Operated as: UA7999
Data from: flightstats.com · All times are displayed in airport local time

Related searches
When is Off Peak LIRR
When is Off Peak
When is Off Season

Anushka Sharma was scared while shooting NH10 – watch ...
www.youtube.com/watch?v=K1z3wEYvrXs ▾
By BollyWood Review · 3 min · 670 views · Added 10/02/2015
Anushka Sharma was scared while shooting NH10 ... After taking a look at the teaser video of NH10 ... Anushka Sharma was scared while shooting NH10 ...

Anushka Sharma & "NH10" Team REACTS on "AIB Knockout ...
www.youtube.com/watch?v=zt0VErCADzQ ▾
By Bollywood Hardcore · 3 min · 29 360 views · Added 06/02/2015
Anushka Sharma & "NH10" Team REACTS on "AIB Knockout Roast Video CONTROVERSY" ... Sonakshi Sinha's Take On AIB Roast ...

NH10 Movie Review, Trailer, & Show timings at Times of India
timesofindia.indiatimes.com › ... › Hindi › Movie Reviews ▾
13/03/2015 · ... click to give your rating/review,Meera and Arjun drive down NH10 ... switch off that; ... Deciding to take a road-trip, they head down NH10 ...

Takin' off all my Clothes off - YouTube
www.youtube.com/watch?v=jJ1SlbE-9Us ▾
By Karate Phone · 4 min · 169 104 views · Added 18/04/2007
... NH and it was also ... Dante and Lulu "I Want To Rip Your Clothes Off" 03/02/10 ... world cup not .the boys take all ther clothes off as they ...

Fundamenta Informaticae XXI (2001) 1001–1016
DOI 10.3233/FI-2012-0000
IOS Press

Natural language understanding and prediction: from formal grammars to large scale machine learning

Nicolae Duta
New England Research and Development Center
Microsoft
niduta@microsoft.com

Abstract. Scientists have long dreamed of creating machines humans could interact with by voice. Although one no longer believes Turing's prophecy that machines will be able to converse like humans in the near future, real progress has been made in the voice and text-based human-machine interaction. This paper is a light introduction and survey of some deployed natural language systems and technologies and their historical evolution. We review two fundamental problems involving natural language: the language prediction problem and the language understanding problem. While describing in detail all these technologies is beyond our scope, we do comment on some aspects less discussed in the literature such as language prediction using huge models and semantic labeling using Marcus contextual grammars.

Keywords: Natural language understanding, language modeling, language prediction

1. Introduction

Scientists have long dreamed of creating machines humans could interact with by voice. In his most cited paper published in 1950, *Computing machinery and intelligence* Turing predicted that "at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted" [46]. "Thinking machines" involve multiple capabilities: recognizing the words which are said, understanding their meaning and being able to produce a meaningful reaction (e.g, answer a question which may imply reasoning in addition to simply querying a fact database, perform an action/transaction, etc).

Although, after several decades of research, one no longer believes Turing's prophecy that machines will be able to converse like humans in the near future, real progress has been made in the voice and

Address for correspondence: Microsoft NERD Center, One Memorial Drive, Cambridge, MA, 02142, USA

text-based human-machine interaction. From a theoretical viewpoint, modern computational linguistics started in the late 1950s when Noam Chomsky introduced the theory of generative grammars which aimed at producing a set of rules that correctly predict which combinations of words form grammatical sentences [15]. The first practical attempts at natural language understanding by a computer were at MIT and Stanford in the 1960s: Daniel Bobrow's STUDENT system which used natural language input to solve algebra word problems and Joseph Weizenbaum's ELIZA system that could carry a dialog on many topics (although it did not have a real understanding of the language)[1].

However, the early systems were only using written text input; it would take two more decades of research until Automatic Speech Recognition (ASR) could allow for voice input. Throughout 1990-2000s, the Defense Advanced Research Projects Agency (DARPA) in the United States conducted several programs to advance the state of the art in ASR, spoken dialog and information extraction from automatically recognized text (ATIS 1990-1994, Hub4 1995-1999, Communicator 1999-2002, Ears 2002-2005, Gale 2005-2010) [52]. These scientific advances have also been sustained by the introduction and exponential growth of the *World Wide Web* and by the huge increase in computing power and miniaturization that led to the today's proliferation of smartphones.

This paper is a light introduction and survey of some of the deployed natural language systems and technologies and their historical evolution. We review two fundamental problems involving natural language: the language prediction problem and the language understanding problem. While describing in detail all these technologies is beyond our scope, we do comment on some aspects less discussed in the literature such as language prediction using huge models and semantic labeling using Marcus contextual grammars.

2. Natural Language Prediction

Language prediction is defined as the ability to predict which words naturally follow a given word sequence. It is generally assumed that natural languages are governed by a probability distribution on word sequences and the language prediction (actually called *Statistical Language Modeling*) models are trying to derive a good estimate of this distribution [4].

Language modeling/prediction has started as a part of the Automated Speech Recognition research effort and is now extensively used in most systems which convert some form of signal into text using a Bayesian approach:

- Automated Speech Recognition (ASR) for acoustic to text mappings [25]
- Optical Character Recognition (OCR) and Handwriting recognition which map document images into text [32][38]
- Automated Machine Translation (AMT) which maps text written in one language into text written in a different language [28]
- Spelling correction systems which map incorrectly spelled text into the correct form [6]
- Word completion and prediction systems (predict following letters in a word or words in a sms/email message considering context and previous user behavior) [9]

[1] A detailed historical perspective can be found in [45]

These systems are all trying to find the text sentence which maximizes the posterior probability $P(Sentence|Signal)$ which according to Bayes rule can be written as

$$P(Sentence|Signal) = P(Signal|Sentence) \times P(Sentence)/P(Signal) \tag{1}$$

$P(Signal|Sentence)$ is the underlying signal model based on acoustic, visual, translational, etc cues while $P(Sentence)$ describes the likelihood of a given sentence in a language.

Since a natural language like English has a lexicon of the order of 10^6 words, it is not possible to directly estimate $P(Sentence)$ for all sentences. Early ASR systems have restricted their language models to the set of sentences appearing in a training set. The drawback was that the system could only output one of the sentences it had seen in training no matter what the user said. Though the system had the ability to reject an input (not produce a text output if $P(Sentence|Signal)$ was too low), that was not very helpful in a practical system that had to deal with unconstrained speech.

Several techniques have been proposed for estimating $P(S)$ for every sentence $S = \{w_1, w_2 ... w_m\}$ (w_i are the sentence words in the order they are spoken) [4][40]; currently the most widely used models are based on decomposing $P(w_1, w_2 ... w_m)$ into a product of conditional probabilities

$$P(w_1, w_2 ... w_m) = P(w_m|w_{m-1} ... w_1) \times P(w_{m-1}|w_{m-2} ... w_1) \times P(w_1). \tag{2}$$

Since it was still impractical to directly estimate $P(w_1, w_2, ..., w_m)$ for long word histories, one assumed that words far away in the history of a target word do not have a large influence. That is, the word sequence $w_1, w_2, ..., w_m$ behaves like a Markov chain of some order n. Therefore one only needs to estimate the statistical distributions of n consecutive word sequences called *n-grams*. According to these models, the probability of a sentence can be decomposed into a product of conditional *n*-gram probabilities. Although counterintuitive, *n*-gram models take no advantage of the syntactic or semantic structure of the sentences they model.

However, if we are using the Maximum Likelihood (ML) estimate for

$$P_{ML}(w_n|w_{n-1}, ..., w_1) = Count(w_n, ..., w_1)/Count(w_{n-1}, ..., w_1) \tag{3}$$

we are facing the issue of assigning a null probability to those *n*-grams not seen in the training data. Even with a training corpus in excess of a few billion words (that's about the size of all newspaper text published in the US in the 1990s) there are still 10-20% valid 3-grams which have not been seen before (last row of Table 1). To properly handle them, one applies a technique called interpolated discounting (also called *smoothing*): set aside a part of the probability mass to account for unseen events and recursively interpolate longer history probabilities with shorter history probabilities:

$$P(w_i|w_{i-1}, w_{i-2}) = P_{ML}(w_i|w_{i-1}, w_{i-2}) \times \alpha(w_i, w_{i-1}, w_{i-2}) + \beta(w_{i-1}, w_{i-2}) \times P(w_i|w_{i-1}) \tag{4}$$

where α and β are called *smoothing functions* and model the amount of the probability mass that is left aside for unseen *n*-grams. To maintain a probability model we need it to sum to 1 over w_i:

$$\beta(w_{i-1}, w_{i-2}) = 1 - \sum_{w_i} P_{ML}(w_i|w_{i-1}, w_{i-2}) \times \alpha(w_i, w_{i-1}, w_{i-2}) \tag{5}$$

A large body of language modeling research in the 1990s has focused on finding suitable values for α and β. Two popular choices are called Witten-Bell [51] and Kneser-Ney [27] discounting:

$$
\begin{array}{lcc}
 & Witten-Bell\ discounting & Kneser-Ney\ discounting \qquad (6) \\
\alpha = & \dfrac{Count(.|w_{i-1},w_{i-2})}{Uniq(.|w_{i-1},w_{i-2})+Count(.|w_{i-1},w_{i-2})} & 1-\dfrac{D(Count(w_i|w_{i-1},w_{i-2}))}{Count(w_i|w_{i-1},w_{i-2})} \qquad (7) \\
\beta = & \dfrac{Uniq(.|w_{i-1},w_{i-2})}{Uniq(.|w_{i-1},w_{i-2})+Count(.|w_{i-1},w_{i-2})} & \dfrac{\sum\limits_{w_i} D(Count(w_i|w_{i-1},w_{i-2}))}{Count(w_i|w_{i-1},w_{i-2})} \qquad (8)
\end{array}
$$

Starting in the early 2000s, the proliferation of documents posted on the internet generated a potentially huge LM training set. However, internet scraped text could not be directly used for LM training since: (i) Almost all of it was out of domain for the systems built at the time[2]. (ii) The computational resources (memory and computing speed) were not sufficient to accommodate the huge number of resulting *n*-grams (a 5 billion-word newspaper corpus generates about 0.8B unique 3-grams and 1.5B unique 4-grams).

Multiple directions of research started to address these issues. One of them was LM pruning: some *n*-grams considered not too informative were discarded (although after being used in computing global statistics of the data). The simplest pruning technique is to discard the least frequent, higher-order *n*-grams which one may assume are not statistically significant. A more sophisticated technique is entropy pruning which considers the relative entropy between the original and the pruned model [44]. However, there appears to be a complex interaction between the pruning method/parameters and the type of discounting used in training the model and that can impact the speech recognition accuracy by as much as 10% [12].

A second research direction was to redesign the LM estimation toolkits and speech recognition pipelines to accommodate all *n*-grams seen in the data. It is important to know the difference between *n*-grams that are unobserved because they are rare and those that are impossible[3]. As shown in Table 1, keeping one billion 3-4 grams in the LM reduces the Word Error Rate (WER) by about 6% in the Broadcast News recognition domain [20]. Although received with skepticism in the academia [11], this direction (along with a distributed data processing framework like Map-Reduce [16]) largely contributed to the recent success of the Google ASR system [13].

For many speech recognition applications (e.g. conversational speech) sufficient in-domain language data has not always been available and a solution was found to be the use additional out-of-domain data (especially internet scraped). Unfortunately, a simple mix of two (different in nature) corpora does not usually result in a better LM and a successful mixing strategy is often regarded as an art. Therefore, a third area of research has focused on combining in-domain with out-of-domain data or even bootstrapping an in-domain LM only using out-of-domain data [8][19].

While this is still an active research area we would like to point out two interesting phenomena. The first is that the colloquial forms of some languages like Arabic and their literary counterparts (e.g. the Modern Standard Arabic-MSA used in newspaper articles and TV broadcasts) although have the same

[2] *n-gram* models are very sensitive to changes in the style, topic or genre of the text on which they are trained (called in-domain data) [40].

[3] An analysis of the text currently used in sms messages and twitter postings shows that almost everything is now possible due to word mispelling, abbreviation and lack of syntactic structure

Table 1. The effects of LM prunning on the English broadcast news task [20]

.

$LMOrder$	$LMsize$	$HitRates$	WER
	[4-grams,3-grams]	[4-grams,3-grams]	
3	[0, 36M]	[0, 76%]	12.6%
3	[0, 305M]	[0, 84%]	12.1%
4	[40M, 36M]	[49%, 76%]	12.1%
4	[710M, 305M]	[61%, 84%]	11.8%

word lexicon, share very few of the higher order n-grams (see Table 2). That means that published texts and TV transcripts are not effective for training a conversational LM [26].

Table 2. Vocabulary coverage and 3-gram hit rates for LMs based on the Arabic Conversational (150K words), Broadcast News (300M words) and Conversational + BN data

LM training data	Vocabulary coverage	3-gram Hit Rate
Conversational alone	90.6%	20%
Broadcast News	89.5%	4%
Conversational + News	96.6%	21%

The second phenomenon is that even though there may still be a significant accuracy gap between speech recognition using a fully in-domain LM and that using a bootstrapped LM, the semantics of the recognized sentence may be far less impacted. That is, one can still figure out the semantic intent of a sentence even when some of the words are misrecognized [19][47].

Finally, we would like to mention the latest trends in Language Modeling. *Discriminative language models* (DLMs) [14] aim at directly optimizing word error rate by rewarding features that appear in low error hypotheses and penalizing features in misrecognized hypotheses. Since the estimation of discriminative LMs is computationally more intensive than regular n-gram LM one has to use distributed learning algorithms and supporting parallel computing infrastructure [16]. *Neural network language models* embed words in a continuous space in which probability estimation is performed using neural networks (feed-forward or recurrent, very recent work is based on multiple hidden layer networks called *deep networks* [2]). The expectation is that, with proper training of the word embedding, words that are semantically or gramatically related will be mapped to similar locations in the continuous space. Because the probability estimates are smooth functions of the continuous word representations, a small change in the features results in a small change in the probability estimation and NNLM may achieve better generalization for unseen n-grams.

3. Natural Language Understanding

3.1. Brief history

During the last couple of decades there has been a tremendous growth of deployed voice driven language understanding systems; however mostly designed for limited domains. At first, these systems were able to recognize and interpret (through fixed grammars) some predetermined phrases and named entities like locations or business names. Most popular were the Directory Assistance (DA) systems built by TellMe, Phonetic Systems/Nuance, BBN, Jingle, Google, etc.

Later on, the ASR technology started to support constrained digit sequences (dates, phone numbers, credit card and bank account numbers) and form filling directed dialog systems were designed for tasks like flight reservation. In such systems, users are asked to provide answers to what the system has asked for, which often consists of a single piece of semantic information. Directed dialog systems evolved into mixed-initiative systems where both users and the system can control the dialog flow and which allowed users to provide more semantic information in a single utterance and in any sequence they choose. The language understanding task became higher resolution with more semantic entities in need to be identified, segmented and normalized.

The DARPA Airline Travel Information System (ATIS) project [3] was initiated in the 1990s for the flight information domain. Users provide some flight attributes like departure and destination cities, dates, etc. However there were no constraints on how the information could be expressed. That is, users could say either "I need a flight reservation from Boston to Miami leaving tomorrow and returning in two weeks" or "Please show me the flight to Miami departing Boston tomorrow". One can notice that beyond this freedom of expression there is a clear semantic structure with crisp and unambiguous semantic entities like Departure/Arrival Cities/Date/Times. These entities, known as "semantic slots" or "frame elements" are considered to be part of a set of templates (semantic frames) which represent the structure of the semantic space. The language understanding component in a frame-based system has to choose the correct semantic frame for an utterance and to segment and normalize the associated semantic slots. For example, the "Departure Date" slot expressed as the word "tomorrow" has to be normalized to something like "03/11/2013" in order to be useful for searching a flight database. Most ATIS systems employed either a statistical classification approach (those coming from the speech processing community) such as AT&T's CHRONUS [37] and BBN's hidden understanding models [30] or a knowledge-based approach (mostly from the computational linguistics community) such as the MIT's TINA [42], CMU's Phoenix [50], and SRI's Gemini [17].

TINA [42] is basically a context-free grammar converted to a probabilistic network and implements a seamless interface between syntax and semantics. The initially bootstrapped context-free grammar is built from a set of training sentences where each sentence is translated by hand into a list of the rules invoked to parse it. The rule set is converted to a form that merges common elements on the right-hand side (RHS) of all rules sharing the same left-hand side (LHS). Elements on the LHS become parent nodes in a family tree. Through example sentences, they acquire knowledge of who their children are and how they can interconnect. The injection of domain-dependent semantics is done by replacing the low-level syntactic non-terminals with semantic non-terminals. For example, the syntactic rule-based derivation

SUBJECT => NOUN_PHRASE => ARTICLE NOUN => the Hyatt

is replaced by the semantic derivation [48]

SUBJECT => ARTICLE PLACE => ARTICLE HOTEL => the Hyatt

A main limitation of the knowledge-based systems is that the grammar design process is tedious, slow and requires a lot of expertise. The semantic space partition into semantic frames may be subjective and the set of slots for a frame are imposed in a top-down fashion rather than extracted from data. Therefore some natural language sentences may not be well modeled in this framework.

At the other end of the semantic spectrum are systems which only need to extract the sentence intent without other semantic entities. An example of such systems are the *Call Routers* whose goal is to automatically route a telephone query from a customer to the appropriate set of agents based on a brief spoken description of the problem. Call routers are nowadays deployed in most of the large call centers because they reduce queue time and call duration, thus saving money and improving customer satisfaction by promptly connecting the customer to the right service representative. These systems remove the constraints on what a user can say but at the expense of limiting the target semantic space. That is, call routers are specifically built for business verticals (e.g. telecommunication, government, utility companies) and are only designed to detect the kinds of semantic intents specific to that vertical (e.g. a telecommunication provider may allow a customer to perform one of several actions: canceling some service, resolving a billing issue, paying a bill, adding a new telephone line, etc).

Well known call routing systems are the AT&T How may I help you? (HMIHY) [22] and the BBN Call director [33]. The users are greeted by an open-ended prompt like How May I Help You?, which encourages them to speak naturally. To find the meaning of a human utterance in a call routing system, the caller's speech is first translated into a text string by an ASR system and the text is then fed into a NLU component called Router. The NLU task is modeled as a statistical classification problem: the text corresponding to an utterance is assigned to one or more of a set of predefined user intents (routes).

3.2. Current NLU architecture

The explosion of mobile computing power that came with the smartphones allowed the development of more sophisticated NLU systems that could handle combinations of many user intents along with the associated named entity extraction. There is now a proliferation of more complex, dialog-based voice search systems and mobile personal assistants that are configured to understand and perform several tasks [18]. Each task may have different sets of semantic entities that can be formulated and uttered differently by different users. The NLU goal in such systems is to also identify which task the user would like to perform (usually called *user intent*).

A modern client-server voice-based transactional system including dialog is depicted in Fig. 1 (see also [49], [23]). A user opens a client application on his phone and utters a sentence (e.g. query or command). The client sends the acoustic signal to the server system where it is first converted into text by the ASR module. Next, a NLU module extracts the semantic information from this text. A popular approach is top-down hierarchical meaning extraction. A semantic domain classifier can be used to determine which part of the semantic space a query belongs to. For example, the query "I need a table for two at the closest Bertucci's restaurant for tomorrow" belongs to the "Restaurant" domain. Then, using domain dependent models, a second classifier finds the query intent (what the user asks for). In our example, the intent is "Restaurant reservation". Finally using domain and intent dependent models, one segments the semantic slots (basic semantic entities) associated with the given domain and intent which have been specified by the user. In our case, the following slots appear and can be extracted from the sentence: (i) Restaurant name = "Bertucci's" (ii) Reservation date = "tomorrow", (iii) Party size = "two" and (iv) Restaurant location = "closest". After that, a normalizer translates each slot value into a form that

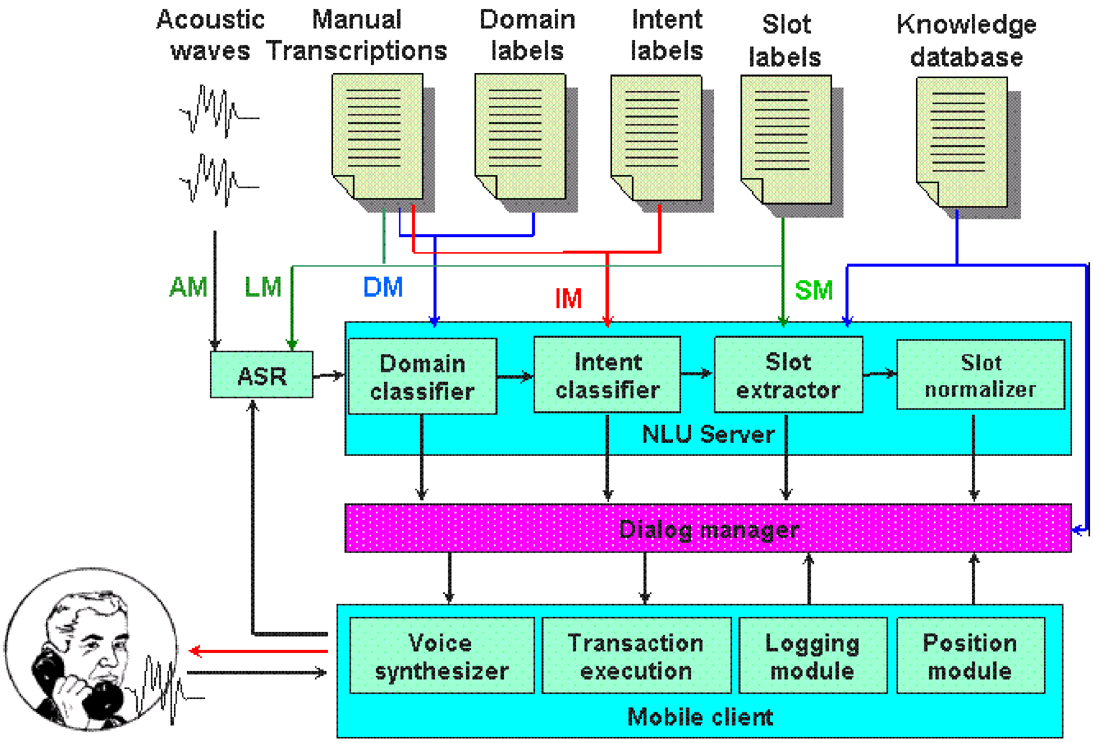


Figure 1. The architecture of a client-server voice-based transactional system including dialog.

can be used to query a knowledge-database. For our query, we could get the following slot normalized values: (i) Restaurant name = “Bertucci’s pizzeria” (ii) Reservation date = “03/15/2013” and (iii) Party size = “2”. The query domain, intent and normalized slot values are further sent out to a *dialog manager* which has detailed information about the domain and determines the next system action. In our case, the dialog manager asks the client application to provide the current user location[4], then it interrogates a business database to find the closest Bertucci’s restaurant and finally detects that a restaurant reservation also requires time information in order to be fulfilled. Therefore, it will issue back to the user a question regarding the reservation time. The dialog manager produces the question as text, but that is fed into a speech synthesizer and turned into an audio signal which is played back to the user. Let’s assume the user answers “Hum let’s say six in the evening”. The NLU system now detects a single semantic slot Time = “six in the evening” which is normalized as Time = “6 pm” and sent to the dialog manager along with “Unknown” domain and intent. The dialog manager, which also keeps track of the dialog states (possible using a stack), knows that this is the missing piece of information from a previous query and it can now take action on the query. Some systems send back to the user the parsed information asking for confirmation: “Ok, I’ll make a reservation for two at Bertucci’s on Main street for March 15th 2013 at 6pm. Is that correct?” If the user agrees, the Execution unit sends all the information to a restaurant reservation service/web site which performs the actual reservation.

One can easily notice that the dialog system architecture in Fig. 1 generalizes all systems built in the

[4] either from the internal GPS or from the wireless provider signal triangulation

past. The DA systems had no client application (at the time users were using landlines), dialog manager (there was a single-shot query which was automatically routed to a human agent if the system returned a low confidence), no domain or intent classifiers (the system's goal was only to return the phone number of a certain business or individual). They only had a primitive slot extractor (either business name or location, though often they were asked for separately) and normalizer. The directed dialog systems added a dialog manager and a small number of fixed intents often specified as a single piece of information[5]. On the other hand, the call routers added an intent classifier (with a number of intents ranging from a few tens to a few hundreds) and a very small number of slots.

From a linguistic viewpoint, these systems could be characterized by the following four criteria (see Table 3 and [49]): the naturalness and the size of the space of input sentences, the resolution of the target semantic representation and the size of the target semantic space. The systems have evolved from a low naturalness, input space, semantic resolution and space size (directed dialog) to medium-high naturalness, large input space, high semantic resolution and space size (today's voice transaction systems).

Table 3. Comparison of several NLU systems with respect to the characteristics of the input utterance space and the output semantic space (adapted from [49])

NLU system	User input	utterances	Target semantic	representation
	Naturalness	Input space	Resolution	Semantic space
Directed dialog	Low	Small	Low	Small
DA	Low	Large	Low	Small
Mixed initiative	Low-medium	Small	High	Small
Call routing	High	Medium	Low	Small
Voice search & personal assistant	Medium-high	Large	High	Large

One important phenomenon is that the text which modern systems are attempting to understand obeys less and less the syntax rules of the language. Spoken language often contains dysfluencies, restarts and recognition errors while written text may be highly abbreviated and/or truncated. For example, an SMS line may be "he like u" while a speech recognized sentence may be "by movie uh kings speech" (the spoken query was "buy movie king's speech").

3.3. Semantic data annotation

As previously mentioned, modern NLU systems often consist of sets of text classifiers which extract various types of semantic information: query domain, query intent, semantic slots and/or other attributes of the domain (facets) which sometimes may only be mentioned implicitly (e.g. the fragment "which make me laugh" in the query "find movies which make me laugh" should be interpreted as a movie genre and one may need some sort of logic reasoning for extracting these mappings [10]).

[5] The system could have asked: "What transaction would you like to perform: Flight information, reservation, cancellation, other?"

These classifiers need to be trained on large amounts of data in which the semantic entities of interest are manually annotated. As shown on top of Fig. 1 several sets of manual annotations are necessary: (i) Speech transcriptions (a textual form of the user spoken utterances) (ii) Semantic domain and/or intent annotations and (iii) Semantic slot annotations. Although one tries to carefully annotate the data, the references produced by different human annotators are not identical. Sometimes that is due to annotator fatigue but most of the time there is a subjective component especially for the semantic annotations. The inter-annotator disagreement may be 6% for speech transcription [21] but it can get much higher when semantics is involved.

Therefore one may try to automate parts of the data annotation process. Semantic slot annotation could be done using Marcus contextual grammars [29][36] which have been theoretically studied for a long time (a brief introduction is given in the Appendix). We will show here an example of how to construct and use such a grammar. Let's assume the vocabulary V is the set of English words and the starting language A over V is the set of sentences a human might use for interacting with an NLU system as described in Section 3.2. The set of selectors correspond to the semantic entities we would like to label and the set of contexts contain the English word we would like to label them with. Let's say S_1 is the set of restaurant names, S_2 is the set of location names and C_1, C_2 are their corresponding semantic labels:

$$S_1 = \{McDonald's, Boston\ \ Market, ...\}, \quad C_1 = \{(< Restaurant >, < /Restaurant >)\},$$
$$S_2 = \{Boston, Cambridge, ...\}, \quad C_2 = \{(< Location >, < /Location >)\}$$

and so on. A possible derivation in this grammar is

Find me a McDonald's in Boston =>
Find me a <*Restaurant*>McDonald's</*Restaurant*> in Boston =>
Find me a <*Restaurant*>McDonald's</*Restaurant*> in <*Location*>Boston</*Location*>

In order to generate correct annotations, we require the derivations to be in maximal global mode. That is, at each derivation step, the word selector x is maximal with respect to all selectors $S_1, ..., S_n$. That is enforced if we always label first the longest semantic entity that could be labeled. The resulting annotated sentence obeys Occam's razor[6] (annotates as many words as possible with as few labels as possible) and is most of the time correct. A simple example is

Find me a Boston Market in Cambridge =>
Find me a <*Restaurant*>Boston Market</*Restaurant*> in Cambridge =>
Find me a <*Restaurant*>Boston Market</*Restaurant*> in <*Location*>Cambridge</*Location*>

If the derivation is not in the maximum global mode one could get:

Find me a Boston Market in Cambridge =>
Find me a <*Location*>Boston</*Location*> Market in Cambridge =>
Find me a <*Location*>Boston</*Location*> Market in <*Location*>Cambridge</*Location*>

which is obviously incorrect.

In [29], the finite and regular families of selectors are investigated. Although in practical systems the vocabulary, starting axioms, selectors and contexts are all finite, the case where the selectors are generated by a context sensitive mechanism is of high interest. That is because one name entity may belong to multiple semantic classes. For example the word "Eagles" belongs to MusicBand, SportsTeam

[6] For a mathematically formalized version of Occam's razor see Ray Solomonoff's theory of universal inductive inference [43]

and Bird classes. For such ambiguous cases, the set of selectors must also contain some contextual words to disambiguate the semantic class. As such, the word "Eagles" should appear in fragments like "Eagles songs" in MusicBand, "Eagles scores" in SportsTeam and "Eagle food" in Bird. The problem becomes even harder when the sets of context sensitive selectors have to be automatically extracted from un-annotated data.

An easy way to implement semantic annotation with a Marcus contextual grammar is by using Finite State Transducer technology [3] [31]. In the Xerox FST toolkit language [3], the grammar shown above can be written as:

```
define Location [{Boston}|{Cambridge}] EndTag(Location); # Selector Location
define Restaurant [{McDonald's}|{Boston Market}] EndTag(Restaurant); # Selector Restaurant
regex Location | Restaurant; # Grammar definition with implicit vocabulary
```

and the annotated output produced by the toolkit is:

```
fst[1]: pmatch Find me a Boston Market in Cambridge
Find me a <Restaurant>Boston Market</Restaurant> in <Location>Cambridge</Location>
```

The main advantage of parsing with FSTs is that the models are very compact (the FST network built using a location list of 320K items is about 25MB in size) and the amount of processing time is very low (a few ms per sentence).

3.4. Semantic classification

Semantic classification is the task of mapping relevant pieces of information from a sentence into semantic labels (classes). It mostly relies on constructing features to represent the sentence and building a classification model. As shown in Fig. 1, one can perform several types of semantic classification. The semantic domain and intent classification assign to each sentence a single class while the semantic slot extraction identifies and labels parts of the sentence[7]. There are mainly two types of statistical classification approaches [41]:

- Generative (also known as Informative) methods that directly model each of the class densities separately. Classification is done by examining the likelihood of each class producing the features ($P(Class|Features) \sim P(Features|Class) \times P(Class)$) and assigning to the most likely class. Although not difficult to train, these methods often lag in accuracy. Some examples are Fisher Discriminant Analysis, Hidden Markov Models and Naive Bayes and they were used in the BBN Call Director [33] and the AT&T HMIHY [22].

- Discriminative methods that model the class boundaries or class membership directly rather than the class feature distributions. Because these models take into account all classes simultaneously, they are harder to train, often involve iterative algorithms and might not scale well. Examples include Neural Networks, Support Vector Machines, AdaBoost (AT&T SLU system [23]), Conditional Random Fields (Microsoft NLU [10]).

In early systems, the feature set used to represent a sentence was mostly a *bag of words* or *n*-grams. Since many words express no semantics, this was later refined to consist of *salient phrases* computed

[7] Slot extraction also performs sentence segmentation and is a more difficult classification task

based on mutual information. For example, the fragment "cents a minute" strongly suggests a calling plan [22]. However, sometimes sentence fragments may have a completely different meaning than any of their constituent words (e.g. "flying spaghetti monster" is a religous sect). The matter is even more complicated if the semantic segmentation is unknown. If "George Washington" is segmented as a single semantic entity then it can be interpreted as a person (US president) name. But if contains two semantic entities then it should be interpreted as a "Town State" entity[8]. In order to address these issues, newer systems include semantic parsing based features [10]. Given a semantic dictionary list (also known as a *gazeteer*), the entity types of the sentence fragments found in the dictionary can be used as features instead of the bare words. Other types of features are Language Model scores, syntactic parsing labels or even semantic class information from another (possibly noisy) source.

3.5. Parsing-based semantics

There has been a large amount of recent work (especially from the *Information Extraction* community) dealing with extracting semantics from queries people submit to search engines. These queries can be either spoken or typed and have been mentioned in Section 3.1 as "voice search" data. One can roughly divide them into [7]:

(i) *Navigational*: reaching a website explicitly requested (e.g. "go to facebook") or a certain state in the dialog flow (e.g. "go back" or "cancel"),

(ii) *Informational*: finding information on the web (e.g. "capital grille restaurant reviews") and

(iii) *Transactional*: conducting a transaction on a website (e.g. "make a reservation at capital grille").

These queries are relatively short, contain many named entities and are often formulated as a concatenation of keywords rather than in a natural language. This particular structure makes it easy to generate a compact representation called *query templates*. A template is a sequence of terms that are either text tokens or variables that can be substituted from some dictionary or taxonomy [5]. For example, if the named entities are replaced by their type in the annotated sentence in Section 3.3 we obtain the template

Find me a <*Restaurant*> in <*Location*>

It has been reported that a large number of queries follow a small number of structured patterns / templates: 90% for real estate and hotel related queries and 80% for automobile and car rental queries [1]. The template extraction process is based on abstracting the semantic entities/slots and sometimes needs a context sensitive mechanism for disambiguation (see the *"Eagles"* example in Section 3.3). There are also queries which are inherently ambiguous (have multiple meanings). For example *"jobs at apple"* may refer to either employment with Apple or to the former Apple executive Steve Jobs [1].

Each parse can receive a score indicative of its quality. While several scoring functions are analyzed in [34], a simple heuristic can be Occam's razor: models which are shorter (contain a smaller number of slots) and more complete (abstract as much as possible of the query) are to be prefered. Notice that this heuristic is in itself an optimization process and it is applied to each query at runtime. This contrasts to the statistical classification methods presented in Section 3.4 where some optimization is performed on a training set and one hopes that it will generalize to unseen samples.

Parsing-based semantics employs a set of query templates and several fact databases to extract the

[8] There exists indeed a town named George in the Washington state and a person could say "George Washington" with the same meaning as "Seattle Washington". However, people usually avoid this kind of ambiguities in their communication.

query intent and semantic slots. The semantic domain and intent classes can be associated to each template rather than to individual queries (a template represents an equivalence class of queries in the semantic space). If *click-through* data (search instances that led to clicks on some of the returned links) is available, this assignment can be done automatically [5], otherwise manual assignment can be performed starting with the templates that have the highest recall (cover the largest query classes). The names of the parsed semantic slots can also be used as features for statistical classification complementing those described in Section 3.4.

There are several advantages of representing queries by template models:

1. Templates generalize the set of target queries and model queries that follow the same patterns but did not appear in the training data [5]. That is especially useful when bootstrapping an NLU system with very little usage data available.
2. Templates models do not require retraining as new entities emerge. If a new restaurant opens, its name can be added to the Restaurant list and all requests applying to other restaurants will generalize to the new one [5].
3. Since they are derived from real data, templates are more comprehensive than hand-crafted rules and far more compact than non-generalizing whitelists (lists of cached queries) [1][5].
4. Template models allow for quick query parsing and matching using FST technology (see Section 3.3 and [34]).
5. Template models do not require an apriori domain schema that specifies the semantic slots and their values. Instead, it learns the most frequent slots automatically while identifying the most relevant templates [1].

Finally we would like to mention that automatically extracted templates have been successfully used for semantic reasoning and relation extraction [35]. A small set of manually identified seed facts that are in a "hidden relation" (e.g. *(Vincenzo Bellini, 1801)*) was used to extract patterns from a large amount of web documents. An example template is "**LHS** *BE_BORN MONTH* **RHS**" (**LHS** and **RHS** denote the Left Hand Side and Right Hand Side of the seed facts respectively). These templates were in turn used to infer the same relationship for many other instances of the two semantic entities (*Person* and *BirthYear*).

4. Conclusions and future developments

After five decades of research, natural language understanding and prediction technology has become an essential part of many human-machine interaction systems (and even human-to-human; see automated translation). We believe that the tipping point for the large scale deployment of this technology has been attained with the introduction of smartphones in the late 2000s. There are now voice-based personal assistants, search and transactional systems for most smartphone platforms [18]. The technology is pushed even further by the search engines (Google, Bing and Yahoo!) which have evolved from simple keyword search to semantic search [24]. They can now provide direct answers to a wide range of questions (e.g. "What's the weather tonight in Boston" or "What are the latest Bruins scores") rather than links to web documents.

5. Appendix

In this section we provide definitions for some acronyms and measures used throughout the text:

WER: Word Error Rate measures the quality of the output produced by a speech recognizer and has typically been measured against a human-made ground truth reference of the audio input. WER is computed as the sum of the errors in each of three classes (word substitutions, insertions and deletions) and is normalized by the number of reference words.

***N*-gram hit rates** express the percentage of *n*-grams in a corpus which are retained (explicitly modeled) by a Language Model.

Marcus contextual grammar is a construct $G = (V, A, (S_1, C_1), ..., (S_n, C_n))$, $n \geq 1$ where V is a vocabulary, A is a finite language over V, S_1,...,S_n are languages over V and C_1,...,C_n are finite subsets of $V^* \times V^*$ (V^* is the set of all words/strings over V, including the empty one). The elements of A are called axioms (starting words), the sets S_i are called selectors, and the elements of sets C_i, written in the form (u, v), are called contexts.

The direct derivation relation on V^* is defined as $x => y$ *iff* $x = x_1x_2x_3$, $y = x_1ux_2vx_3$, where $x_2 \in S_i$, $(u, v) \in C_i$ for some i, $1 \leq i \leq n$. A derivation is called in **maximum global mode** *if* there are no x'_1,x'_2,$x'_3 \in V^*$ such that $x = x'_1x'_2x'_3$, $x'_2 \in S_j$ for some $1 \leq j \leq n$ and $|x'_1| \leq |x_1|$, $|x'_3| \leq |x_3|$, $|x'_2| > |x_2|$.

Semantic template coverage is the ratio of the number of queries that are instances of the template and the total number of queries.

References

[1] Agarwal, G., Kabra, G., Chang, K. C. C.: Towards rich query interpretation: walking back and forth for mining query templates, *in Proc of the 19th international conference on World Wide Web*, 2010, 1-10.

[2] Arisoy E., Sainath T. N., Kingsbury B., Ramabhadran B.: Deep neural network language models. *In Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, 2012, 20-28.

[3] Beesley K. R., Karttunen L.: *Finite state morphology*, Center for the Study of Language and Information Publication, 2003. FST toolkit can be downloaded at http://www.stanford.edu/ laurik/fsmbook/home.html

[4] Bellegarda J. R.: Statistical language model adaptation: review and perspectives, *Speech Communications*, **42**, 2004, 93-108.

[5] Bortnikov E., Donmez P., Kagian A., Lempel R.: Modeling transactional queries via templates, *in Proc. 34th European conference on Advances in Information Retrieval ECIR 2012*, 2012, 13-24.

[6] Brill E., Moore R. C.: An improved error model for noisy channel spelling correction, *in Proc. of the 38th Annual Meeting on Association for Computational Linguistics*, 2000, 286-293.

[7] Broder, A.: A taxonomy of web search, *In ACM Sigir forum*, **36**(2), 2002, 3-10.

[8] Bulyko I., Ostendorf M., Siu M., Ng T., Stolcke A., Cetin O.: Web resources for language modeling in conversational speech recognition, *ACM Trans. on Speech and Language Processing*, **5**(1), 2007.

[9] Burns M.: Nuance supercharges Swype, adds new keyboard options, XT9 predictive text and Dragon-powered voice input, *TechCrunch.com*, 2012, Available at http://techcrunch.com/2012/06/20/ nuance-supercharges-swype-adds- new-keyboard-options- xt9-predictive-text-and- dragon-powered- voice-input/

[10] Celikyilmaz A., Hakkani-Tur D., Tur G.: Statistical semantic interpretation modeling for spoken language understanding with enriched semantic features, *in Proc of IEEE Workshop on Spoken Language Technologies*, 2012, 216-221.

[11] Charniak E., Gales M.: Personal communication, *EARS RT-04 workshop*, Yorktown Heights, NY, 2004.

[12] Chelba C., Brants T., Neveitt W., Xu P.: Study on interaction between entropy pruning and Kneser-Ney smoothing, *in Proc of Interspeech*, 2242-2245.

[13] Chelba C., Bikel D. M., Shugrina M., Nguyen P., Kumar S.: *Large scale language modeling in automatic speech recognition*, Google technical report, 2012, Available at: http://static.googleusercontent.com/ external_content/untrusted_dlcp/ research.google.com/en/us/ pubs/archive/40491.pdf

[14] Chelba C., Xu P., Pereira F., Richardson T.: Distributed acoustic modeling with back-off n-grams, *in Proc of ICASSP 2012*, 2012, 4129-4132.

[15] Chomsky N.: Three models for the description of language, *IRE Transactions on Information Theory*, **2**, 1956, 113-124.

[16] Dean J., Ghemawat S.: MapReduce: simplified data processing on large clusters, *OSDI'04: Sixth Symposium on Operating System Design and Implementation*, San Francisco, CA, December, 2004.

[17] Dowding J., Gawron J. M., Appelt D., Bear J., Cherny L., Moore R., Moran D.: Gemini: A natural language system for spoken language understanding, *in Proc. ARPA Workshop on Human Language Technology*, Princeton, NJ, Mar. 1993.

[18] Duffy J.: Apple's Siri versus Dragon Go! and Vlingo, *PC Magazine*, 10/6/2011, Available at http://www.pcmag.com/article2/0,2817,2394267,00.asp

[19] Duta N.: Transcription-less call routing using unsupervised language model adaptation, *In Proc. Interspeech 2008*, Brisbane, Australia, September 22-26, 2008.

[20] Duta, N., Schwartz R.: Using a large LM, *EARS technical workshop, Eurospeech 2003*, Martigny, Switzerland.

[21] Duta, N., Schwartz R., Makhoul J.: Analysis of the errors produced by the 2004 BBN speech recognition system in the DARPA EARS evaluations, *IEEE Trans. on Audio, Speech, and Language Processing*, **14**(5), 2006, 1745-1753.

[22] Gorin A. L., Riccardi G., Wright J. H.: How may I help you? *Speech Communication*, **23**(1-2), 1997, 113-127.

[23] Gupta N., Tur G., Hakkani-Tur D., Bangalore S., Riccardi G., Gilbert M.: The AT&T spoken language understanding system, *IEEE Trans. on Audio, Speech, and Language Processing*, **14**(1), 2006, 213 - 222.

[24] Imielinski, T., Signorini, A.: If you ask nicely, I will answer: semantic search and today's search engines, *In Proc. IEEE International Conference on Semantic Computing*, 2009, 184-191.

[25] Jelinek F.: *Statistical methods for speech recognition*, MIT Press, 2001.

[26] Kirchhoff K., Bilmes J., Da, S., Duta N., Egan M., Ji G., He F., Henderson J., Liu D., Noamany M., Schone P., Schwartz R., Vergyri D.: Novel approaches to Arabic speech recognition: Report from the 2002 John-Hopkins summer workshop, *In Proc. ICASSP 2003*, I 344-347.

[27] Kneser R., Ney H.: Improved backing-off for m-gram language modeling, *In Proc. ICASSP 1995*, 181-184.

[28] Lopez, A.: Statistical machine translation, *ACM Computing Surveys*, **40**(3), 2008, 1-49.

[29] Marcus S., Paun G., Martin-Vide C.: Contextual grammars as generative models of natural languages *Computational Linguistics*, **24**(2), 1998, 245-274.

[30] Miller S., Bobrow R., Ingria R., Schwartz R.: Hidden understanding models of natural language, *in Proc. Annual Meeting Association for Computational Linguistics*, Las Cruces, NM, Jun. 1994.

[31] Mohri M., Pereira F. C. N., and Riley M.: The design principles of a weighted finite-state transducer library, *Theoretical Computer Science*, **231**, 2000, 17-32.

[32] Mori S., Nishida H., Yamada H.: *Optical character recognition*, John Wiley and Sons, 1999.

[33] Natarajan P., Prasad R., Suhm B., McCarthy D.: Speech enabled natural language call routing: BBN call director, *in Proc. Int. Conf. Spoken Language Processing*, Denver, CO, Sep. 2002.

[34] Parameswaran, A., Kaushik, R., Arasu, A.: *Efficient parsing-based keyword search over databases*, Technical Report, Stanford University, 2012.

[35] Pasca M., Lin D., Bigham J., Lifchits A., Jain A.: Organizing and searching the World Wide Web of facts - Step one: the one-million fact extraction challenge, *in Proc. of the 21st National Conference on Artificial Intelligence (AAAI-06)*, Boston, Massachusetts, 2006, 1400-1405.

[36] Paun Gh.: *Marcus contextual grammars*, Kluwer Academic Publishers, Dordrecht, Boston, London, 1997.

[37] Pieraccini R., Tzoukermann E., Gorelov Z., Levin E., Lee C., Gauvain JL.: Progress report on the Chronus system: ATIS benchmark results, *In Proc. of the workshop on Speech and Natural Language*, 1992, 67-71.

[38] Plamondon R., Srihari S. N.: On-line and off-line handwriting recognition: a comprehensive survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(1), 2000, 63-84.

[39] Price P. J.: Evaluation of spoken language systems: The ATIS domain, *in Proc. DARPA Workshop on Speech and Natural Language*, Hidden Valley, PA, Jun. 1990.

[40] Rosenfeld, R.: Two decades of statistical language modeling: Where do we go from here? , *Proceedings of the IEEE*, **88**(8), 2000.

[41] Rubinstein, Y. D., Hastie, T.: Discriminative vs. informative learning, *In Proc. Third Int. Conf. on Knowledge Discovery and Data Mining*, 1997, 49-53.

[42] Seneff S.: TINA: A natural language system for spoken language applications, *Computational linguist*, **18**(1), 1992, 61-86.

[43] Solomonoff R.: A formal theory of inductive inference, *Information and Control*, Part I: **7**(1), 1964, 1-22.

[44] Stolcke A.: Entropy-based pruning of backoff language models, *in Proc. DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA, 1998, 270-274.

[45] Tur G., De Mori R. (eds): *Spoken language understanding - Systems for extracting semantic information from speech*, John Wiley and Sons, 2011.

[46] Turing A. M.: Computing machinery and intelligence, *Mind*, **49**, 1950, 433-460.

[47] Wang Y. Y., Acero A., Chelba C.: Is word error rate a good indicator for spoken language understanding accuracy?, *inProc. ARPA HLT Workshop*, St. Thomas, USA, pp. 577-582, 2003.

[48] Wang Y. Y., Deng L, Acero A.: *Semantic frame-based spoken language understanding*, in Tur G., DeMori R. Eds. Spoken Language Understanding, John Wiley and Sons, 2011.

[49] Wang Y. Y., Yu D., Ju Y.C, Acero A.: *Voice search*, in Tur G., DeMori R. Eds. Spoken Language Understanding, John Wiley and Sons, 2011.

[50] Ward W., Issar S.: Recent improvements in the CMU spoken language understanding system, *in Proc. DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA, 1998, 270-274.

[51] Witten, I., Bell, T.: The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression, *IEEE Trans. on Inform. Theory*, **37**(4), 1991, 1085-1094.

[52] The history of automatic speech recognition evaluations at NIST, *Available at http://www.itl.nist.gov/iad/mig/publications/ASRhistory*

Analysis of the Errors Produced by the 2004 BBN Speech Recognition System in the DARPA EARS Evaluations

Nicolae Duta, *Member, IEEE*, Richard Schwartz, and John Makhoul, *Fellow, IEEE*

***Abstract*—This paper aims to quantify the main error types the 2004 BBN speech recognition system made in the broadcast news (BN) and conversational telephone speech (CTS) DARPA EARS evaluations. We show that many of the remaining errors occur in clusters rather than isolated, have specific causes, and differ to some extent between the BN and CTS domains. The correctly recognized words are also clustered and are highly correlated with regions where the system produces a single hypothesized choice per word. A statistical analysis of some well-known error causes (out-of-vocabulary words, word fragments, hesitations, and unlikely language constructs) was performed in order to assess their contribution to the overall word error rate (WER). We conclude with a discussion of the lower bound on the WER introduced by the human annotator disagreement.**

***Index Terms*—Error analysis, speech recognition.**

I. Introduction

OVER THE last decade, the large-vocabulary continuous-speech recognition (LVCSR) systems have become more complex and sophisticated in order to respond to the increased demand for accuracy, speed, and reliability [17]. The technological complexity makes it increasingly difficult to understand the recognition systems' behavior and explain why they are not yet working as well as they should [3], [20]. Nevertheless, there has been a continuous effort to analyze the errors incurred in the automatic speech recognition process.

Greenberg *et al.* [5], [6] performed a thorough analysis of the eight systems present in the NIST 2000 Switchboard Corpus evaluation. They used a 54-min subset of the Switchboard corpus which was phonetically annotated with respect to about 40 acoustic, linguistic, and speaker characteristics. The correlation between those data characteristics and the recognition-error patterns was subsequently probed using decision trees. The authors found that the recognition errors were mostly correlated with the number of phonetic-segment substitutions within a word. That is, the probability of a word being incorrectly recognized increased significantly when more than 1.5 phones were misclassified. It was also shown that the speech rate (measured in syllables per second) was highly correlated with the error patterns as well (see also [12]). Utterances slower than three syllables per second or faster than six syllables per second had 50% more recognition errors than utterances within the normal speaking range. Those correlations were found to be consistent over the eight systems analyzed.

Stolcke and Shriberg [21], [22] looked into how speech disfluencies affected the following word predictability within the Switchboard and ATIS corpora. They showed that the language model (LM) transition probabilities were significantly lower at hesitation transitions and that was attributable to both the target word and the word history. It was also suggested that fluent transitions in sentences with a hesitation elsewhere were significantly more likely to involve unmodeled n-grams than transition in fluent sentences. Based on the findings above, the authors listed disfluencies as "one of the factors contributing to the poor performance of the automatic speech recognizers" although they did not show explicit statistics for how disfluencies correlate with the recognition errors. They also proposed a language model that predicted disfluencies probabilistically and took hidden disfluency events into account. Although the model locally reduced the word perplexity, it had no impact on the recognition accuracy.

A recent analysis of spontaneous speech recognition errors appeared in Furui *et al.* [3]. It was performed on 510 min of spontaneous Japanese speech, and it introduced a regression model for the recognition accuracy as a function of six signal and speaker attributes: average acoustic frame likelihood, speech rate, word perplexity, out-of-vocabulary (OOV) rate, filled pause rate, and repair rate. The authors found that the recognition accuracy was mostly correlated with the repair rate and OOV rate and to a somewhat lesser extent with the speech rate. They hypothesized that the strong effect on errors of the repair and OOV rates was due to the fact that "a single recognition error caused by a repair or an OOV word triggers secondary errors due to linguistic constraints."

Several other studies (see [1] and the references therein) attempted to model the relationships between some features present in the speech signal and the recognition word error rate (WER) using logistic regression. The regression model was subsequently used to predict the correctness of the recognition hypotheses.

Manuscript received September 30, 2005; revised May 10, 2006. This work was supported by the Defense Advanced Research Projects Agency under its EARS Program. The associate editor coordinating the review of this paper and approving it for publication was Dr. Isabel Trancoso.

N. Duta was with the Speech and Language Processing Department, BBN Technologies, Cambridge, MA 02138 USA. He is now with the Natural Language Understanding Group, Nuance Communications, Burlington, MA 01803 USA (e-mail: nicolae.duta@nuance.com).

R. Schwartz and J. Makhoul are with the Speech and Language Processing Department, BBN Technologies, Cambridge, MA 02138 USA (e-mail: schwartz@bbn.com; makhoul@bbn.com).

Digital Object Identifier 10.1109/TASL.2006.878268

Palmer and Ostendorf [18] proposed a technique to explicitly model the errors in the speech recognizer's output in order to improve the name entity recognition performance in an information extraction task. They computed statistics for the name entities occurring in the Hub-4 Topic Detection and Tracking data and reported that "the percentage of name words that are OOV is an order of magnitude larger than words in other phrase categories."

In May 2002, the Defense Advanced Research Projects Agency (DARPA) started a research program called EARS (Effective, Affordable, Reusable, Speech-to-text) whose major goal was to reduce recognition word error rates for conversational telephone speech (CTS) and broadcast news (BN) down to the 5%–10% range, running in real-time on a single processor [23]. Progress made in the recognition of English was measured each year on a "Progress Test" (kept fixed for the duration of the program and undisclosed to the participating sites) as well as on "Current Tests" which changed each year and were made public after the official evaluation. Evaluation conditions became more difficult each year by imposing runtime limits, automatic segmentation requirements, and broadening the data sources. However, due to technological improvements and increasingly more data available for training,[1] the word error rates decreased from around 30% for BN and 50% for CTS to around 10% and 15%, respectively. As noted in [17], the EARS-evaluated systems have achieved "remarkable convergence across both sites and domains," with the top systems showing no statistically significant difference in performance [8], [9].

After the 2003–2004 EARS workshops, we performed detailed analyses of the errors our system made in both BN and CTS English evaluations. Since the correlation between acoustic properties of the speech data and the recognition errors was previously investigated [1], [5], [6], we mainly focused on how the errors were distributed, whether they occurred independently, and whether they were correlated with some language properties of the data. Our analyses show that many of the remaining errors are not random but have rather specific causes, occur in clusters, and differ to some extent between the BN and the CTS domains. The BN system is mostly challenged by the proper nouns in the news stories and by the utterance end-points; the CTS system is challenged by a combination of speech disfluencies, high speech rate, and word contraction; and both systems make substitution errors on short or (acoustically) similar words.

The goal of this paper is to quantify the frequencies of the most common error types as well as the errors' correlation with challenging speech events like OOVs, word fragments, hesitations, and disfluent speech. In Section IV, we propose a method to easily detect regions of very high (99%) recognition accuracy in the system's output, which amount to at least half of the test data. One can resegment the test set in order to keep fixed the high-accuracy regions produced in the first decoding stage. Subsequently, it may be possible to reduce the decoding time as well as to improve the recognition performance by combining with the results produced using the original segmentation. Finally, Section V explores the human annotator disagreement when transcribing the same audio and its impact on how low a WER can be achieved.

II. Systems, Models, and Data Description

The recognition results reported in this paper were obtained using the BBN RT04 (Rich Text) system fully described in [13], [19]. In brief, the system consists of the following.

1) A phoneme decoder-based speech segmenter.
2) 14 Perceptual Linear Prediction (PLP) [7] derived cepstral coefficient and energy front-end.
3) A two-pass decoder with state-tied mixture (STM) [14] acoustic and 2-gram LM models in the first pass and state-clustered tied-mixture (SCTM) [15] noncrossword acoustic and 3-gram LM models in the second pass in a Viterbi beam search, followed by either N-best list (for BN) or lattice (for CTS) rescoring using SCTM cross-word acoustic models and 4-gram LM.
4) A two-stage decoding process; the first decoding stage uses speaker independent (SI) models while the second stage uses speaker adaptively trained (SAT) models. The adaptation process is done using two feature-space transforms (a speaker-specific heteroscedastic linear discriminant analysis HLDA [11] and a constrained maximum likelihood linear regression (CMLLR) transform [4]) and 2–16 model parameter transforms (maximum likelihood linear regression (MLLR) [10]).

Our BN system runs in 10× real time (RT) while the CTS system is 20 × RT.

We performed the error analysis on the BN Eval03 and Eval04 test sets and the CTS Eval01 and Deval04[2] sets, which were made available by NIST following the official DARPA evaluations [8], [9]. A quantitative description of the four data sets along with the BBN's system accuracy on them is shown in Table I. All test sets are transcribed by NIST/LDC and also include annotated tokens for disfluent speech (word fragments and hesitations).

We used recognition lexicons of 61K (BN) and 57K (CTS) unique words, to which the most frequent 3K word pairs were added as compounded words. The OOV rate attained was quite small: 0.15%–0.7% over the four sets. The language models we used in this study contained 737 million 4-grams (BN) and 435 million 3-grams (CTS) and were trained on 0.5–1.5 billion words.

III. Qualitative Error Analysis

A. Error Types Present in Both BN and CTS

The main error type that is shared by BN and CTS is substitution of short or (acoustically) similar words (see Table II for a few examples). These errors make up 15% to 25% of all errors. In such cases it is hard even for humans to distinguish among different choices based on local information only. Parsing the sentence might help in a few BN instances, although often the

[1] More than 2000 h of acoustic training data and over 1 billion words of language training data (although only a fraction of the language training is annotated speech) are now available for both BN and CTS.

[2] CTS Deval04 consists of both CTS Eval03 and Dev04 test sets.

TABLE I
SUMMARY OF THE BN AND CTS TEST SETS ON WHICH WE PERFORMED THE ERROR ANALYSIS

Test set	Words	Reference sentences	Segmented utterances and average utt. length	OOVs	Optional words	WER
BN Eval03	24790	508	1318 (19)	47 (0.2%)	380 (1.5%)	8.3%
BN Eval04	46576	935	2358 (19)	320 (0.7%)	1063 (2.3%)	14.2%
CTS Eval01	62909	5895	5895 (11)	149 (0.24%)	2649 (4.2%)	20.1%
CTS Deval04	113991	25725	12623 (9)	167 (0.15%)	5571 (4.9%)	17.0%

TABLE II
EXAMPLES OF SUBSTITUTION OF SHORT OR SIMILAR WORDS IN BN AND CTS RECOGNITION

Reference	Hypothesis
americans who STRUGGLE to understand	americans who STRUGGLED to understand
the cause of a fire that GUTTED A nearly	the cause of a fire that GOT INTO nearly
airlines with THE background TO that	airlines with A background OF that
israeli troops MORE THAN sixty	israeli troops **** WITHIN sixty
stories that will be NEWS later today	stories that will be USED later today
have you done THIS CALL before	*have you done THESE CALLS before*
*from hawaii *** EVEN your parents ARE born*	*from hawaii AND THEN your parents WERE born*
with all the PERVERSION and stuff	*with all the CONVERSIONS and stuff*

TABLE III
EXAMPLES OF WORD-SPLITTING ERRORS (THE REFERENCE IS SPELLED AS A SINGLE WORD) IN BN AND CTS RECOGNITION

HAND WRITTEN	WASTE LAND	WORK WEEK	ICE BOX
OFFICE HOLDERS	COUNTER INTELLIGENCE	SWAMP LAND	CO STARS
SPY MASTER	SCHOOL TEACHER	AFTER SHOCK	MID DAY
MULTI MILLION	*CHEESE BURGERS*	*NON SMOKING*	*HANG OUT*
UNDER RATED	*OVER CROWDED*	*AUTO PILOT*	*SECOND HAND*
ROLLER BLADING	*BREAST FEEDING*	*BLACK OUT*	*EASY GOING*

information necessary to select the "right" choice may be spread across several sentences.

There are also three common error types which are less frequent but which might be easier to fix than the previous ones.

1) Word splitting (or joining) into valid words accounts for 2%–3% of all errors (e.g., HANGOUT → HANG OUT and HARD WORKING→ HARDWORKING, see Table III for more examples). Although the number of such instances is relatively low, each occurrence generates two errors (a substitution along with a deletion or insertion). Many of these cases should be considered equivalent in scoring and for each such possibility one can replace the system output by the most frequently used version.
2) Plurals are often misrecognized as "⟨word⟩ is" (e.g., CARRIAGES → CARRIAGE IS). Some of these errors might be solved using sentence parsing information in a post-processing step.
3) Errors due to inconsistent spelling (e.g., OKAY → O.K, BOUTROUS → BOUTROS, TRAVELLING → TRAVELING). In many cases, the reference is incorrect and one needs to be more careful about spelling conventions.

B. BN Specific Errors

The error analysis revealed the following BN specific errors.

1) Errors generated by proper nouns (person names or places) account for about 10%–15% of the errors (see Table IV for a list of name errors made on BN Eval04). These are mostly due to insufficient training (especially LM training) or no training at all (OOVs, e.g., IVANISEVITCH). We found that about three quarters of the OOV words are name entities.[3] A misrecognized name is often split up and causes several errors (e.g., BRASWELL → BROWN AS WELL) with an average of 1.5–2 errors per word. If the lexicon contains names acoustically close but with different spellings, the system may output any of related spellings (e.g., HANSSEN → HANSEN or HANSON). The mistaken names are usually different on each test set and the 10–15 most frequently misrecognized names account for one third of all name-related errors. A possible

[3] That is somewhat lower than Palmer's estimate [18]. The remaining OOV words consists of rare words (e.g., ESTRANGEMENTS), common words preceded by a prefix (e.g., PROSLAVERY, REPUBLICATION), or improvised words (e.g., SCALAWAG).

TABLE IV
ERRORS GENERATED BY PROPER NOUNS ON BN EVAL04

Name	OOV	Instances mis-recognized	Errors generated	Correctly recognized	Recognized as
VAN LEW	N	26	63	2	THEN LOU
SCALAWAGS	Y	19	42	0	
MALVO	Y	18	25	0	MALVEAUX
DRU SJODIN	Y	9	24	0	DREW SHOULD DEAN
MUHAMMAD	N	15	18	2	MOHAMMED
IAN HUNTLEY	N	6	16	6	
CHEVAUX	Y	9	14	0	SHOW BOAT, SHOULD VOTE
JAWAD AL AMERI	Y	4	14	0	
ACCUWEATHER	N	5	13	2	
KARACHI	N	11	13	7	
CULLEN	N	10	10	1	COLLIN,COLIN,COLLINS
Other names		272	533		
Total		404	785 (12%)		

TABLE V
BOUNDARY WORD ERROR RATE COMPARED TO THE TOTAL WER

Test set	BN Eval03	BN Eval04	CTS Eval01	CTS Deval04
Boundary errors	300	887	1635	2881
Total errors	2065	6594	12672	19403
Boundary WER	11.4%	19.4%	20.7%	17.7%
Total WER	8.3%	14.2%	20.1%	17.0%

solution to the name problem is a time-adaptive lexicon and LM update using training data from a time period immediately preceding the test data [16]. However, the update data does not usually contain sufficient training for the name context, so some context sharing with the regular training data may be needed.

2) There are more errors toward the utterance end-points than there are in the center (e.g., the BN Eval04 WER on the first and the last utterance words is 19% versus 13% on other words, see Table V). This could be due to a segmentation problem (the automatic segmentation misses the true sentence boundary) or just to having less context in the language model. However, the CTS system does not produce a higher WER on end-points neither on Eval01 (manually segmented) nor on Deval04 (automatically segmented).

C. CTS Specific Errors

We have found the following CTS specific errors.

1) A significant number of errors occur around speech disfluencies: hesitations, repeats, partially spoken words.[4] In such cases, both the acoustic and the language model may be inaccurate; since many word sequences are unique and have never occurred before, they cannot be adequately modeled by the language model. We performed a cheating experiment where the small (60K 3-grams) test set was added to the full language model, and that especially helped in these situations (it halved the unadapted WER). A few examples of disfluency-related errors are shown in Table VI.

2) Deletion of word sequences. There are multiple instances where sequences of two to four consecutive words are deleted from the system's output (Table VII). We listened to the audio for 17 such cases, and almost every time, the deletion could be attributed to a combination of severe word contraction, very high speech rate, and low volume. Moreover, in many such cases, the reference was not accurate; it described what the speaker intended to say rather than what he/she actually said.

IV. QUANTITATIVE ERROR ANALYSIS

A. Error Clustering

The alignments between the reference and the best hypothesis suggested that about two thirds of the errors do not occur in isolation but rather in groups of two to eight errors (see first row of Table X). Therefore, the errors do not appear to be independent, since under an independence assumption more than 70% of the errors should be isolated (according to a binomial distribution over samples of the same length as the test utterances). Since the errors are not homogeneously distributed throughout the test set (there are regions, e.g., speaker turns or even full shows, with a much higher error rate than the average), we decided to test the error clustering hypothesis by computing local statistics like the probability of an error given short histories of correct/wrong recognitions. We show the error versus correct state transition automaton in Fig. 1 ($\langle s \rangle$ corresponds to the beginning of a sentence while $\langle /s \rangle$ is used to mark the sentence end).[5] One can notice the following.

[4] That does not imply that the average WER measured around disfluencies has to be higher than the overall WER. Many disfluencies may produce no errors, while others may be very costly. We show a quantitative analysis in Section IV-C.

[5] We only show the transition probabilities for the BN Eval04 and the CTS Deval04 sets. The figures corresponding to the remaining two sets are very similar in each domain and were omitted for space reasons. The transition probabilities were computed under the assumption that the hesitation tokens were NOT optional, fact which slightly increased P(Err) for the CTS domain.

TABLE VI
EXAMPLES OF HESITATION-RELATED ERRORS IN CTS RECOGNITION (HYP DENOTES THE REAL SYSTEM OUTPUT; HYPC IS THE OUTPUT OF THE CHEATING EXPERIMENT)

Ref:	like (%HESITATION) the whole HEAVEN'S gate thing WAS IT HEAVEN'S gate I CAN NOT REALLY (-ember)
Hyp:	like ON the whole HEAVENS gate thing FROM THE HEAVENS gate * THING YOU KNOW
HypC:	like ON the whole heaven's gate thing was it heaven's gate * THING YOU KNOW
Ref:	she had a **** HAUNTED HOUSE (%hesitation) there was a BELL that would ***** RING AT a certain
Hyp:	she had a HARD TO HAVE %hesitation there was a BELLOW that would BRING IT TO a certain
HypC:	she had a haunted house AND there was a bell that would ring at a certain

TABLE VII
EXAMPLES OF WORD-SEQUENCE DELETIONS BY THE CTS SYSTEM

Ref:	to (%hesitation) YOU KNOW all my friends are getting married and EVERY ONE IS HAVING babies
Hyp:	to *** **** all my friends are getting married and ***** *** ** ****** babies
Ref:	maybe AT a higher stage OF DEVELOPMENT THAN we are
Hyp:	maybe ** a higher stage ** *********** THAT we are
Ref:	secretary of state IN THE HOSPITAL in THE hospital after undergoing a serious surgery
Hyp:	secretary of state ** *** ******** in A hospital after undergoing a serious surgery

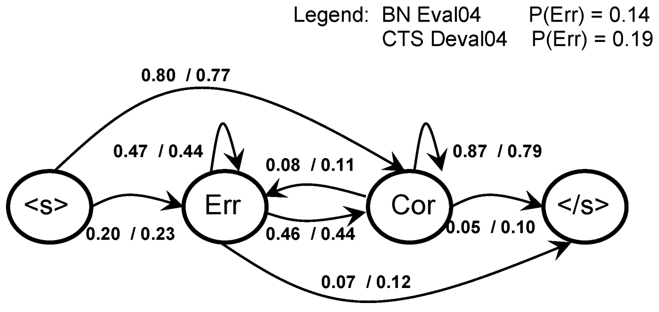


Fig. 1. Transition probabilities between error and correct states for the BN and CTS systems.

1) $\mathrm{P(Err|Err)} > 2.5 * \mathrm{P(Err)}$ for both domains, which shows that it is a lot more likely for an error to follow another error than to occur independently of the history.
2) $\mathrm{P(Err|\langle s \rangle)}$ and $\mathrm{P(Err|\langle /s \rangle)}$[6] (corresponding to errors made on the utterance end-points) are 50% higher than P(Err) for BN, which verifies our direct measurements in Table IV.

A similar automaton corresponding to groups of three adjacent words is shown in Fig. 2 ("0" in a state denotes an error, while "1" denotes a correct word, e.g., "000" represents three consecutive errors). The error clustering trend appears very strong: $\mathrm{P(Err|Err, Err, Err)}$ is 2.5 to 3 times higher than P(Err), and even when the history contains a correct word, one still has a much increased probability of error. As expected, the correctly recognized words are also strongly clustered. However, as long as the most recent word is correct, the remaining history does not matter anymore: $\mathrm{P(Cor|Cor, Cor, Cor)} = \mathrm{P(Cor|Cor, Err, Cor)} = \mathrm{P(Cor|Cor)} = \mathrm{P(Cor)} = 1 - \mathrm{P(Err)}$. That is, for correctly recognized words, the third-order Markov model is reduced to a first-order model, while for errors it is still a third-order model.

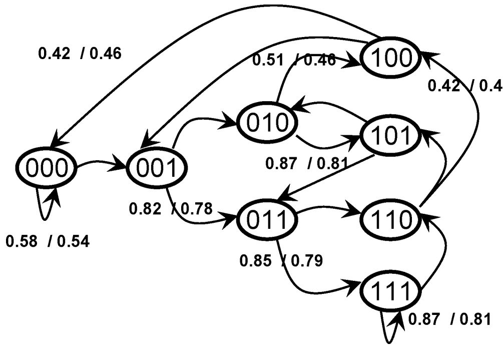


Fig. 2. Transition probabilities for a three word-class (either "correct" or "error") state automaton. A "0" in the state denotes an error output word while a "1" denotes a correct word. Each transition arc is labeled by the probability of observing the right-most word class (either 0 or 1) of the target state given the source state (e.g., the transition from [110] to [101] is labeled by $\mathrm{P}(1|011) = \mathrm{P}(\mathrm{correct}|\mathrm{error}, \mathrm{correct}, \mathrm{correct})$).

B. Identifying Clusters of Correctly Recognized Words

Most state-of-the-art LVCSR systems employ some statistical measure to assess the confidence in the system's output. In this section, we propose a simple method for estimating regions of correctly recognized output.

For each test set, we aligned the list of the 100 best hypotheses, and we analyzed the regions that only had a single choice for each word. Fig. 3 shows a single word choice versus multiple word choice automaton computed using the hypotheses generated after the second (speaker adapted) decoding stage. This automaton has a clustering trend similar to that in Fig. 1 on both BN and CTS systems and all four test sets. Given that we are in a single choice region, the probability to remain there is 0.66 while the overall probability of a single choice word

[6] According to Bayes' law, $\mathrm{P(Err|\langle /s \rangle)} = \mathrm{P(\langle /s \rangle|Err)} * \mathrm{P(Err)}/\mathrm{P(\langle /s \rangle)}$ $= \mathrm{P(\langle /s \rangle|Err)} * \mathrm{P(Err)}/[\mathrm{P(\langle /s \rangle|Err)} * \mathrm{P(Err)} + \mathrm{P(\langle /s \rangle|Cor)} * \mathrm{P(Cor)}]$ $= 0.07 * 0.14/[0.07 * 0.14 + 0.05 * 0.86] = 0.19$.

TABLE VIII
RECOGNITION STATISTICS ON THE OPTIONAL TOKENS (HESITATIONS AND WORD FRAGMENTS)

Test set	Optional tokens	Hesitations/word fragments deleted	Optional full words recognized	Optional full words deleted	Optional tokens substituted
BN Eval03	380 (1.5%)	348 (92%)	0	0	32 (8%)
BN Eval04	1063 (2.3%)	876 (82%)	8	9	164 (18%)
CTS Eval01	2649 (4.2%)	1888 (71%)	34 (1.5%)	32 (1.5%)	680 (26%)
CTS Deval04	5571 (4.9%)	4234 (76%)	85 (1.7%)	185 (3.3%)	1057 (19%)

TABLE IX
RECOGNITION STATISTICS ON THE UNLIKELY LANGUAGE CONSTRUCTS

Test set	Unlikely constructs	Correctly recognized	Misrecognized	Total damage (errors)
BN Eval03	1065 (4.3%)	862 (81%)	203 (19%)	392
BN Eval04	2301 (5.0%)	1755 (76%)	546 (24%)	1228
CTS Eval01	2312 (3.7%)	1801 (78%)	511 (22%)	1228
CTS Deval04	4082 (3.6%)	3201 (78%)	881 (22%)	2044

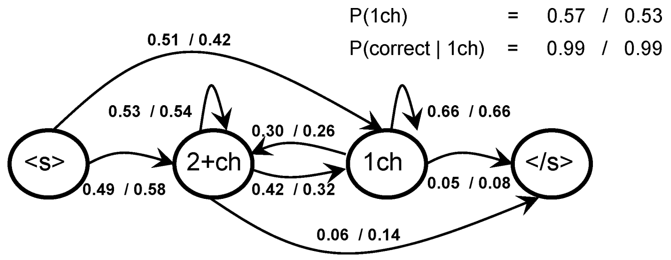


Fig. 3. Transition probabilities between single-choice and multiple-choice states for the BN and CTS systems following the second (speaker adapted) decoding stage.

is only 0.55. At the same time, $\mathrm{P(Cor|single\ choice)} = 0.99$. In other words, there is 99% recognition accuracy on the single-choice region (about 55% of all words) of each test set.

Similar results are obtained if the hypotheses generated after the unadapted decoding are used in computing the transition probabilities. The only difference is that P(single choice) is slightly lower: 0.53 for BN and 0.48 for CTS. That is, the regions of high recognition confidence are smaller when the unadapted system output is used (compared to the output generated by the adapted system). We noticed that most of the 1% errors found in the single choice per word regions using the unadapted hypotheses are not fixed after adapted decoding.

If the test set is resegmented at the boundaries of the single word choice regions, it is possible that redecoding only the multiple word choice regions in the subsequent adaptation stages could help in two ways: 1) speed-up the system and 2) improve accuracy by allowing system combination with the results obtained using the original (unadapted) segmentation.

C. Impact of Nonfluent and Nonmodeled Speech on Errors

We have also measured how nonmodeled words (OOVs), word fragments, unintelligible speech (generically marked as "%hesitation" by the human annotators), as well as other forms of nonfluent speech (repeats, fillers, edits)[7] influence the WER. One should first note that the reference tokens marked as word fragments and unintelligible speech are optionally deletable for scoring purposes. That is, one introduces an error if such a token is substituted but not if it is deleted. All optional tokens are considered when computing the total number of reference words by which one normalizes the WER.

As shown in Table VIII, about 1.5%–2.5% (BN) and 4%–5% (CTS) of all reference words are marked as optional, and they are a lot more frequent in CTS than in BN. Very few ($<6\%$) of the optional tokens are actually full words which can be correctly recognized. According to Columns 4–5 of Table VIII, 30%–50% of them are indeed correctly recognized. All other tokens are either nonmodeled word fragments or generic hesitations. Both BN and CTS systems are tuned to delete 70%–90% of the optional tokens in order not to introduce errors. As a consequence, especially our CTS system, avoids producing output on some high rate speech regions and on partially spoken words although those words are not marked as optional in the reference. That agrees with our observation in Section III-C that the CTS system is unbalanced toward deletions.

The recognition statistics for the unlikely language constructs are shown in Table IX. Since we used very large language models, a word pair (word, word history) was not explicitly modeled only 3.5%–5% of the time. In most (75%–80%) of these nonmodeled cases, the system still produced the correct output. However, the mistakes due to unlikely language constructs are very costly: each misrecognized word generates multiple errors (see last column in Table IX).

Table X shows the (per cluster) distribution of the errors generated by the three event types: OOVs, optional tokens, and unlikely language constructs. The count of each event (and its associated error count) was computed for each error cluster length

[7] These events were not explicitly marked as such in our references. However, we considered them to occur in regions that did not contain OOVs but in which our very large LMs had to be backed-off up to a unigram. Given that our LM's bigram hit rate is 98% on fluent (like newspaper) text, there is only a 2% chance that a fluent word pair is not modeled; most remaining pairs are examples of unlikely language constructs.

TABLE X

STATISTICS OF THE ERROR DISTRIBUTION (COUNTS OF THE ERRORS OCCURRING IN CLUSTERS/GROUPS OF LENGTH i) ALONG WITH ERROR CONTRIBUTION FROM OOVS, OPTIONAL TOKENS, AND UNLIKELY LANGUAGE EVENTS. THE FIRST FIGURE IS THE EVENT COUNT, THE SECOND IS THE ASSOCIATED ERROR COUNT (e.g., SECOND COLUMN IN ROW OOVS SHOWS THAT 254 OF THE 1744 ERRORS THAT OCCUR IN GROUPS OF TWO ARE GENERATED BY 131 OOV WORDS FOUND IN 127 TWO-ERROR CLUSTERS)

BN Eval04	**Counts of [target events, errors associated] present in clusters of length i:**								
Target events	1	2	3	4	5	6	7	8	**Total**
All errors	2035	1744	1077	600	315	228	112	112	6594
OOVs	72/72	131/254	60/153	29/92	17/65	6/36	2/14	0/0	320/714
Optional tokens	91/91	31/60	20/51	14/24	0/0	1/6	6/7	0/0	164/248
Unlikely language	182/182	163/308	86/231	45/148	34/130	16/84	10/56	4/32	546/1228

CTS Deval04	**Counts of [target events, errors associated] present in clusters of length i:**								
Target events	1	2	3	4	5	6	7	8	**Total**
All errors	6613	4994	2886	1916	1100	696	371	232	19255
OOVs	38/38	55/110	21/60	20/80	9/45	9/48	2/14	1/8	167/473
Optional tokens	611/611	232/408	100/222	54/148	39/105	13/42	2/14	2/16	1057/1068
Unlikely language	337/337	251/498	116/348	78/300	49/225	27/138	7/42	3/24	881/2044

TABLE XI

STATISTICS OF THE LANGUAGE MODEL EVALUATION ORDER MEASURED ON THE SYSTEM'S OUTPUT (1-BEST HYPOTHESIS) FOR THE ERROR SAND CORRECT REGIONS AS WELL AS ON OOVS, OPTIONAL TOKENS, AND UNLIKELY LANGUAGE CONSTRUCTS. THE EVALUATION ORDER MEASURED ON THE REFERENCE FOR THE ERROR REGIONS IS ALSO SHOWN FOR COMPARISON

Target events	**LM evaluation order (BN Eval04)**					**LM evaluation order (CTS Deval04)**			
	0	1	2	3	4	0	1	2	3
All error words (reference)	4%	12%	23%	23%	31%	9%	6%	12%	73%
All error words (1-best hypothesis)		9%	32%	30%	29%		2%	11%	87%
Correct words		3%	18%	31%	48%		1%	5%	94%
OOV words		24%	37%	23%	16%		11%	17%	72%
Optional token errors		6%	30%	30%	34%		1%	10%	89%
Unlikely language errors		24%	37%	22%	17%		6%	17%	77%

i (i =1 to 8). For example, on BN Eval04, 72 isolated errors were generated by OOVs, while 131 OOVs occurred in 127 error clusters of length 2 and therefore produced 254 errors. After manually inspecting the error clusters, it appears that for small values of i (2 to about 4) all the errors in a cluster in which one of the three target events mentioned above occurs, can be attributed to that target event.[8] According to Table X, the unlikely language constructs produce the most damage (2.5 errors per occurrence), followed by OOVs (two errors per occurrence) and by optional tokens (1.5 errors per occurrence). This result confirms the hypothesis in Furui *et al.* [3].

It is also interesting to consider the language model behavior on the error and correct clusters as well as on the three event classes mentioned previously. Before measuring this behavior, we have intuitively assumed that whenever a higher order {3–4} n-gram was not modeled by the LM, the recognition system had to consider a shorter history and back-off the probability until the (target, history) was actually modeled. Tables IX and XI show that is the case most of the time. However, when a word is not modeled by the LM and about 8%–15% of the cases the pair (word, immediate history) is not modeled, the system prefers to use higher order n-grams which acoustically resemble the utterance. That is, instead of using the correct 1-gram, the system uses an incorrect {3–4}-gram. In such cases, a whole neighborhood of the target word is misrecognized and multiple errors are generated. That explains why errors due to OOVs and unlikely language are so costly and often occur in 2–4 word clusters.

V. DISCUSSION: ERROR MEASUREMENT

The automatic speech recognition errors are defined by the disagreement between the output of the automatic system and the output of the human recognition (typically called ground truth reference) on the same speech data. We would like to conclude the paper with a discussion of the error rate dependence on the human-made ground truth.

The error measure, called word error rate, is computed as the sum of the errors in each of the three classes (substitutions, insertions, and deletions) and is normalized by the number of reference words. Usually, a single manually generated and

[8] We noticed that the long error clusters (some of which span the entire utterance) can rather be attributed to low-quality (very fast, low volume, accented, noisy) speech, so one can consider them outliers.

carefully annotated (by two independent transcribers with the disagreements adjudicated by a third person) reference is used as a ground truth. Although transcriptions are done carefully, the references produced by different transcriber teams are not identical.

We present our attempt at quantifying and explaining the annotation differences (for a full statistical analysis see [2]). In 2003, BBN contracted WordWave to transcribe 1700 h of Fisher data [9] to be distributed to the EARS community for CTS acoustic training. In order to measure the quality of the "quick" transcriptions, WordWave was asked to transcribe the CTS Eval03 test set for which a careful transcription was provided by MSU-LDC. After alignment, the WordWave transcription showed 11.5% WER w.r.t. to the MSU-LDC transcription. We randomly picked and listened to 15 of the 144 5-min speaker turns which had multiple transcription differences (343 out of 2511 words) and found the following.

1) In about 30% of the cases, the MSU transcription appeared to be correct, some of the differences may have been due to carelessness or fatigue of the WordWave transcriber.
2) In about 15% of the cases, the WordWave transcription appeared to be correct, we noticed a few differences on words with foreign origin (e.g., "LA RUE GASTRONOMI QUE") as well as some cases where MSU transcribed what the speaker intended to say rather than what he/she actually said.
3) In about 25% of the cases, we could not tell which transcription was correct; much of the speech was not audible and there was true ambiguity in the utterance.
4) About 25% of the cases were different spelling conventions (e.g., UH versus AH).
5) About 10% of the differences are due to incomplete annotations of NOISE or LAUGHTER which each transcriber may mark somewhat randomly if the audio is noisy.

After normalizing the spelling conventions and eliminating the NOISE markings, the real differences between the two transcriptions were around 6%–7% (this figure was later confirmed in [2] on multiple transcription sets). As the speech-to-text WER will soon approach the differences among transcribers, we will have to account for these differences when computing the WER. To overcome this problem, several alternative error measures were introduced in [2].

VI. Conclusion

In this paper, we quantified the main error types still present in a speech recognizer's output and measured their correlation with some language properties of the data. We showed that there are both common and specific error types in BN and CTS. However, the main error types are somewhat different.

1) In comparison with BN data, CTS data contains very few name entities, and even though each name still causes more than one error when misrecognized, the total number of name-related errors is small.
2) The disproportionate percentage of errors that occur at the utterance end points in BN did not occur for CTS. It is unclear at this point whether that is due to the test set segmentation or to a weak LM at sentence boundaries.
3) The large percentage of deletions that occur in CTS shows that the system is tuned to avoid errors in regions of disfluent speech a significant number of which are marked as optional. In this way, the average WER around disfluencies does not become higher than the average WER. However, some disfluencies may generate multiple errors (see Tables VI and X).

The four test sets analyzed were consistent with respect to the error types and frequencies. The only exception was the misrecognition of proper names, which was very much dependent on the time period when the test set was collected. Finally, the error analysis shows that many of the remaining errors are not random but have rather specific causes. The challenge is now how to use this information to reduce the WER. That might be possible by designing different solutions for different error classes, and the detection of possible error, or correct regions might aid in this error class specific process.

References

[1] S. Choularton. Investigating the Acoustic Sources of Speech Recognition Errors. [Online] Available: http://www.ics.mq.edu.au/~stephenc/inter2005.pdf

[2] J. Fiscus and R. Schwartz, "Analysis of scoring and reference transcription ambiguity," presented at the *EARS 2004 Meeting*, Palisades, NY, Nov. 7–10, 2004.

[3] S. Furui, M. Nakamura, T. Ichiba, and K. Iwano, "Why is the recognition of spontaneous speech so hard?," in *Proc. 8th Int. Conf. Text, Speech, Dialogue*, Karlovy Vary, Czech Republic, 2005, pp. 9–22.

[4] M. J. F. Gales, "Maximum-likelihood linear transformation for HMM-based speech recognition," *Comput. Speech Lang.*, vol. 12, pp. 75–98, 1998.

[5] S. Greenberg, S. Chang, and J. Hollenback, "An introduction to the diagnostic evaluation of the Switchboard-corpus automatic speech recognition systems," in *Proc. NIST Speech Transcription Workshop*, College Park, MD, May 16–19, 2000, [Online] Available: http://www.nist.gov/speech/publications/tw00/pdf/cp2110.pdf..

[6] S. Greenberg and S. Chang, "Linguistic dissection of switchboard-corpus automatic speech recognition systems," in *Proc. ISCA Workshop on Automatic Speech Recognition: Challenges for the New Millennium*, Paris, France, 2000, pp. 195–202.

[7] H. Hermansky, "Perceptual linear predictive PLP analysis for speech," *J. Acout. Soc. Amer.*, vol. 4, pp. 1738–1752, 1990.

[8] A. Le. Rich transcription 2003 spring speech-to-text evaluation results. presented at *EARS 2003 Meeting*. [Online] Available: http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/rt03s-stt-results-v9.pdf

[9] ——, 2004 fall rich transcription speech-to-text evaluation. presented at *EARS 2004 Meeting*. [Online] Available: http://www.nist.gov/speech/tests/rt/rt2004/fall/rt04f-stt-results-v6b.pdf

[10] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density HMMs," *Comput. Speech Lang.*, vol. 9, pp. 171–186, 1995.

[11] S. Matsoukas and R. Schwartz, "Improved speaker adaptation using speaker dependent feature projections," in *Proc. IEEE ASRU Workshop*, St. Thomas, U.S. Virgin Islands, 2003, pp. 273–278.

[12] N. Mirghafori, E. Fosler, and N. Morgan, "Fast speakers in large vocabulary continuous speech recognition: Analysis and antidotes," in *Proc. Eurospeech Conf.*, Madrid, Spain, 1995, pp. 491–494.

[13] L. Nguyen, B. Xiang, M. Afify, S. Abdou, S. Matsoukas, R. Schwartz, and J. Makhoul, "The BBN RT04 English broadcast news transcription system," in *Proc. Interspeech Conf.*, Lisbon, Portugal, 2005, pp. 1673–1676.

[14] L. Nguyen and R. Schwartz, "Single-tree method for grammar-directed search," in *Proc. ICASSP Conf.*, Phoenix, AZ, 1999, pp. 613–616.

[15] ——, "Efficient 2-pass N-best decoder," in *Proc. Eurospeech Conf.*, vol. I, Rhodos, Greece, 1997, pp. 167–170.

[16] K. Ohtsuki, N. Hiroshima, M. Oku, and A. Imamura, "Unsupervised vocabulary expansion for automatic transcription of broadcast news," in *Proc. ICASSP Conf.*, vol. I, Philadelphia, PA, 2005, pp. 1021–1024.

[17] M. Ostendorf, E. Shriberg, and A. Stolcke, "Human language technology: Opportunities and challenges," in *Proc. ICASSP Conf.*, vol. V, Philadelphia, PA, 2005, pp. 949–953.

[18] D. Palmer and M. Ostendorf, "Improving information extraction by modeling errors in speech recognizer output," in *Proc. Human Language Technology Workshop*, San Diego, CA, 2001, pp. 1–5.

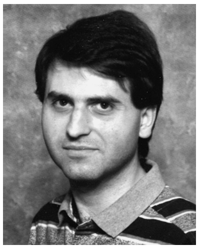
[19] R. Prasad, S. Matsoukas, C. Kao, J. Ma, D. Xu, T. Colthurst, O. Kimball, R. Schwartz, J.-L. Gauvain, L. Lamel, H. Schwenk, G. Adda, and F. Lefevre, "The 2004 BBN/LIMSI $20 \times$ RT English conversational telephone speech recognition system," in *Proc. Interspeech Conf.*, Lisbon, Portugal, 2005, pp. 1645–1648.

[20] E. Shriberg, "Spontaneous speech: How people really talk, and why engineers should care," in *Proc. Interspeech Conf.*, Lisbon, Portugal, 2005, pp. 1781–1784.

[21] E. Shriberg and A. Stolcke, "Word predictability after hesitations: A corpus-based study," in *Proc. Int. Conf. Spoken Language Processing*, Philadelphia, PA, 1996, pp. 1868–1871.

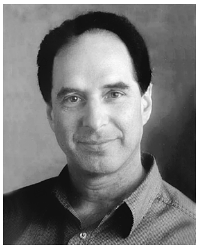
[22] A. Stolcke and E. Shriberg, "Statistical language modeling for speech disfluencies," in *Proc. ICASSP*, Atlanta, GA, 1996, pp. 405–408.

[23] C. Wayne, "Effective, affordable, reusable, speech-to-text," presented at the *EARS 2003 Meeting*, Boston, MA, May 19–22, 2003.



Nicolae Duta (M'91) received the B.S. degree in applied mathematics from the University of Bucharest, Bucharest, Romania, in 1991, the D.E.A. degree in statistics from the University of Paris-Sud, Paris, France, in 1992, the M.S. degree in computer science from the University of Iowa, Iowa City, in 1996, and the Ph.D. degree in computer science and engineering from Michigan State University, East Lansing, in 2000.

He is currently a Scientist in the Natural Language Understanding Group, Nuance Communications, Burlington, MA. From 2000 to 2005 he was a Scientist in the Speech and Language Processing department at BBN Technologies, Cambridge, MA. He also held temporary research positions at INRIA-Rocquecourt, France, in 1993 and Siemens Corporate Research, Princeton, NJ, from 1997 to 1999. His current research interests include computer vision, pattern recognition, language understanding, automatic translation, and machine and biological learning.



Richard Schwartz received the S.B. degree in electrical engineering from the Massachusetts Institute of Technology, Cambridge.

He joined BBN Technologies, Cambridge, MA, in 1972 and is currently a Principal Scientist. He specializes in speech recognition, speech synthesis, speech coding, speech enhancement in noise, speaker identification and verification, machine translation, and character recognition.

John Makhoul (F'80) received the B.E. degree from the American University of Beirut, Beirut, Lebanon, the M.Sc. degree from the Ohio State University, Columbus, and the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, all in electrical engineering.

Since 1970, he has been with BBN Technologies, Cambridge, where he is a Chief Scientist working on various aspects of speech and language processing, including speech recognition, optical character recognition, language understanding, speech-to-speech translation, and human–machine interaction using voice. He is also an Adjunct Professor at Northeastern University, Boston, MA.

Dr. Makhoul has received several IEEE awards, including the IEEE Third Millennium Medal. He is a Fellow of the Acoustical Society of America.