WEBTALK: TOWARDS AUTOMATICALLY BUILDING INTERACTIVE SYSTEMS THROUGH MINING WEBSITES

Speaker: Junlan Feng

Outline

- Part I:

  WebTalk: Towards Automatically Building Spoken Dialog Systems Through Mining Websites

- Part II:

  Question Answering with Discriminative Learning Algorithms

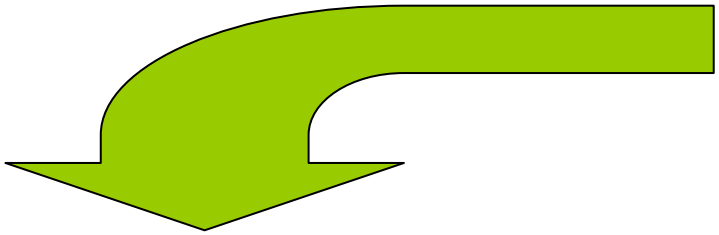
WEBTALK

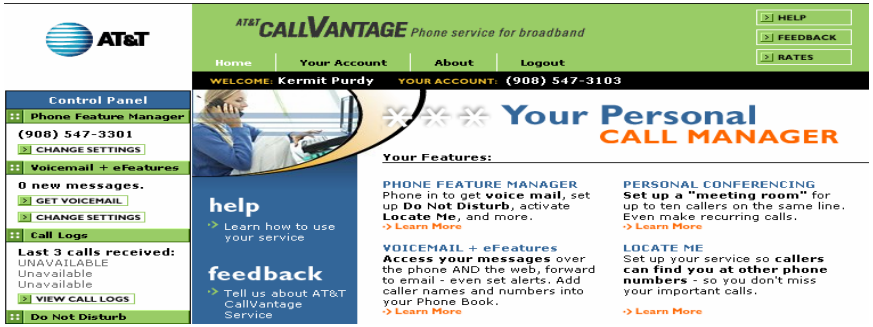
- ***Goal:***
  - *Automatically constructing text and speech-based dialog systems from company websites*
  - *Customer Care Question Answering System*
- ***Advantages:***
  - *Automation: Constructing dialog applications with zero human intervention*
  - *Synchronization: Synchronizing with updates on the website.*
  - *Complexity: Possessing as much knowledge as the website contains.*

Challenges & Applications

- ***Challenges:***
  - *Robust Website understanding*
  - *Webpage-based language model that reflects human-machine interaction*
  - *Web data driven dialog manger*
  - *Spoken Language understanding that requires no hand rules and annotation*
  - *Language Generation that's grammatically correct without rules*
- ***Applications:***
  - *Telephone-based Customer Care Service*
  - *Internet-based Customer Care Service*

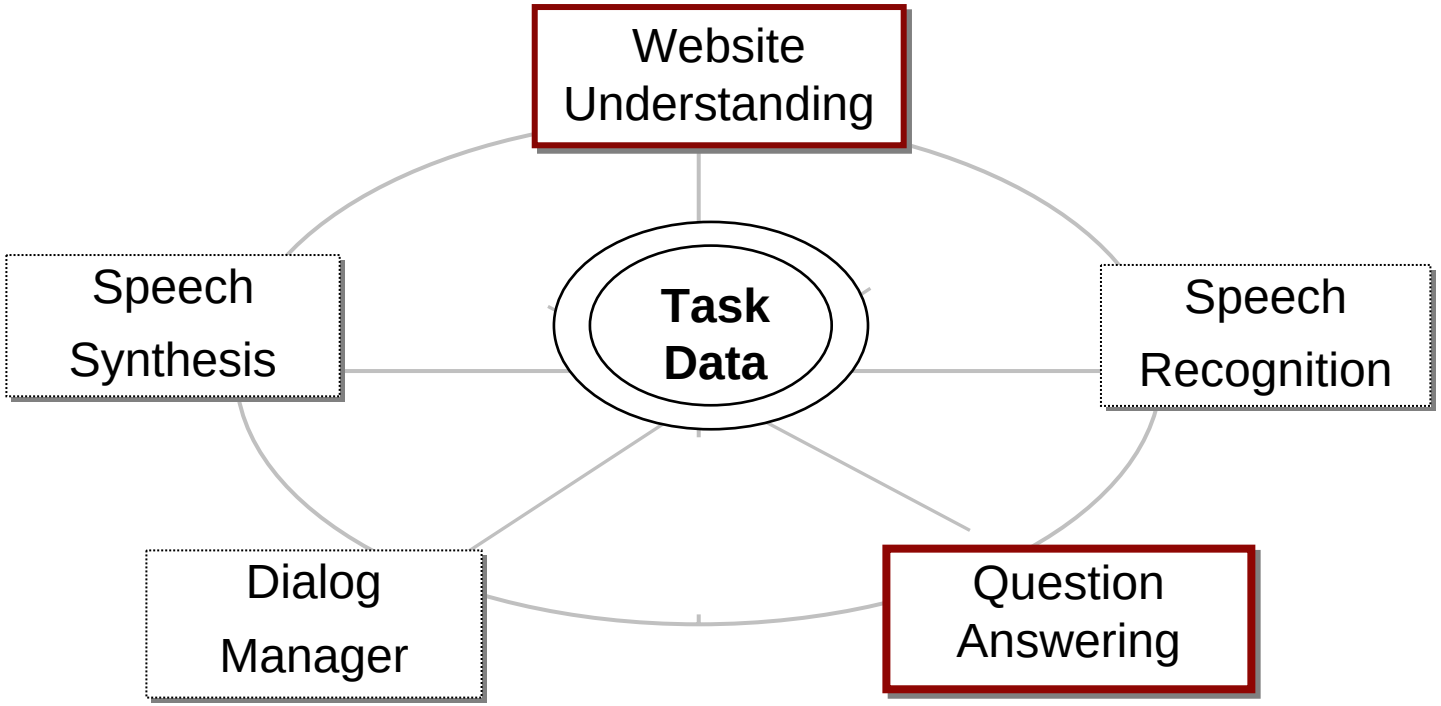
An example



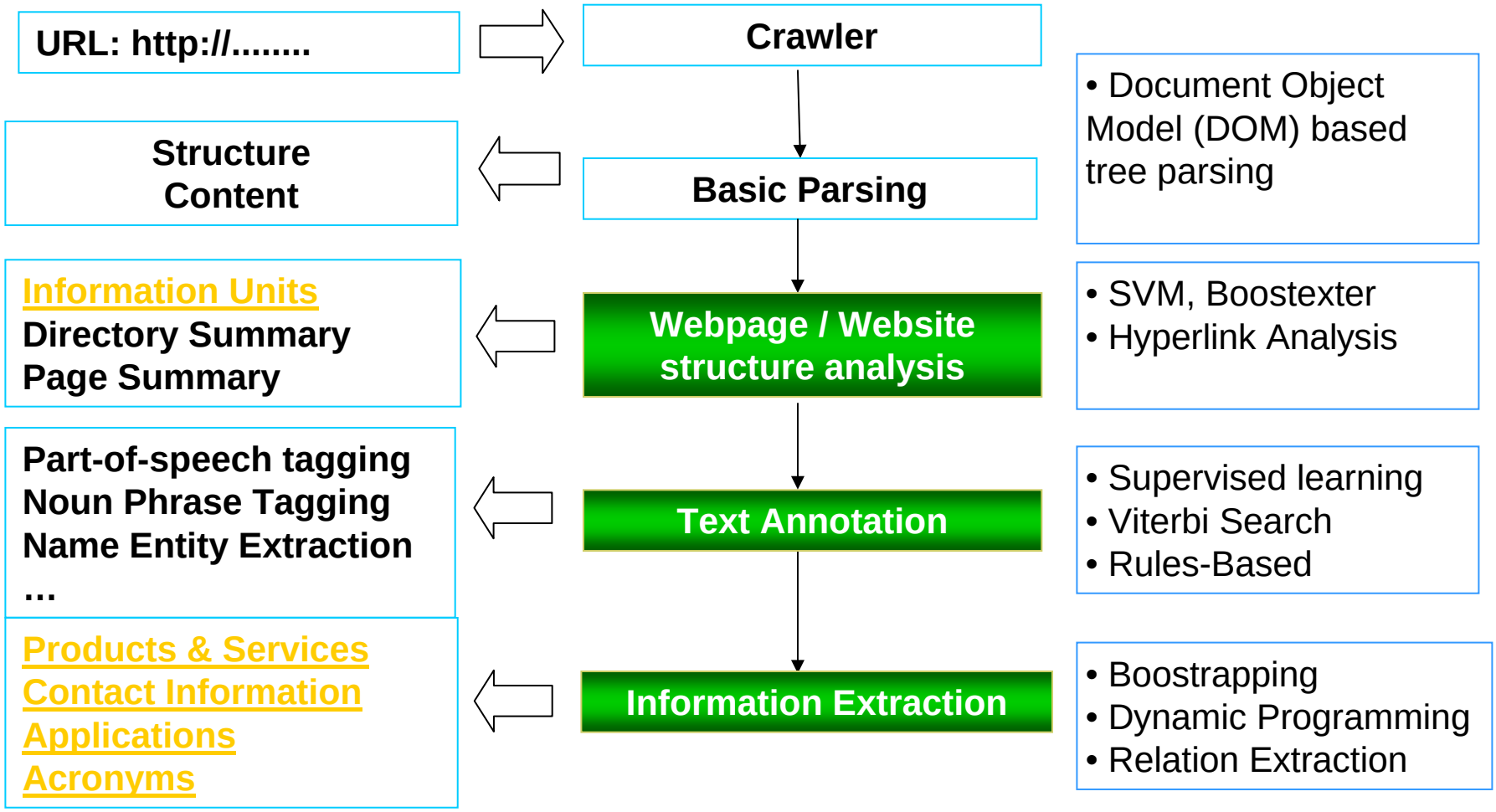


W: Hi, this is AT&T Callvantage service - welcome . How may I help you?
U: I have a question.
W: OK, What is your question?
U: Can I keep my callvantage phone number, when I move to a new address?
W: You can still take your AT&T CallVantage Service and phone number with you, as long as you have a high speed cable modem or DSL Internet connection.
U: What's your current rate for calling China?
W: Call Destination : CHINA - BEJING
Per Minute Rate : To LandLine : $0.06
To Mobile : $0.09
U: Thanks
W: You're welcome. Anything else?

A diagram of major components in Webtalk



Website Understanding



Webpage Structure Analysis

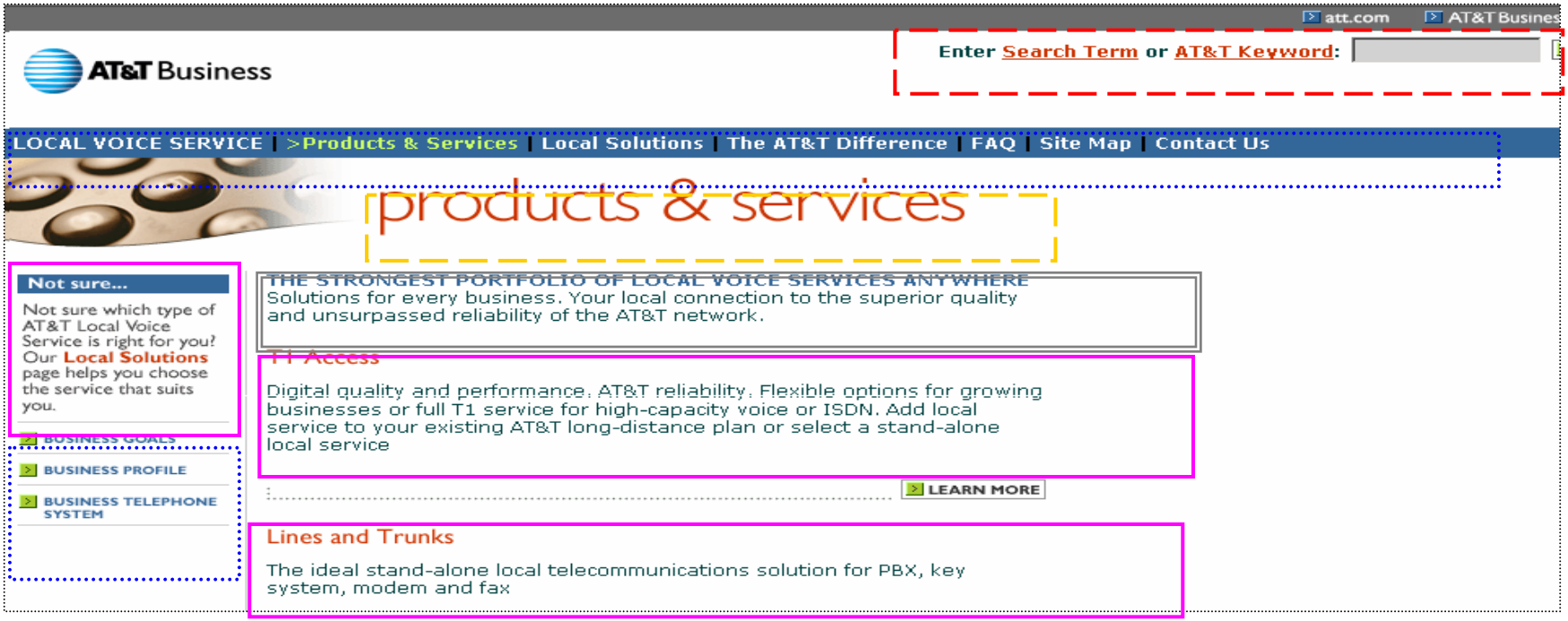


Figure 1: An example web page, where

-Form -Menu -Page-Title - Normal-Content -Heading-Content

Speech Recognition (ASR)

- ***ASR Engine:***
  - *AT&T Watson speech recognizer*
- ***Acoustic Models:***
  - *The acoustic model was trained using utterances collected from other deployed spoken dialog services*
- ***Language Models:***
  - *Task:*
    - Train a statistical language model using text data on a given website
  - *Challenges:*
    - The web language is significantly different than conversational utterances that are typically observed in a spoken dialog system.

Speech Recognition (continued)

- *Language Model:*
  - *Approaches:*
    - In order to take advantage of the website content, we translate the web sentences provided by the Web Page Parser into conversational style utterances using the following three steps:
      - **Filtering:** remove the common task independent sentences from the web text.
      - **Predicate/argument extraction:** semantically parse the web sentences, using the ASSERT tool from the University of Colorado, and extract the predicate/argument pairs.
      - **Stitching:** insert the predicate and arguments to the corresponding slots in the conversational templates (manually written or previously learned), which are sequences like: I would like to <PRED> <ARG>.
    - Generated New utterances are then merged with data collected from other applications to create an n-gram language model.

Question Answering

Customer Question:	Question Preprocessing	Query Formulation	Answer Search	Answer Filter	System Response
"How much does CDS cost?"	POS-Tagging: [..CDS//NNP] Question Classification: [PRICE] Product & Service Locating: [CDS]	Extract Terms: [CDS, cost] Extract Essential Terms: [CDS] Query Extension [CDS-> Consumer Digital service, Cost->price]	•IR •Topic Match •Phrase-Match •NP Match •ET Match •Matched Position •Proximity Search •Length •Spelling Errors	Question Answer Type match [price→ [$..]] Product and Service Matching	Answer / Help / Rejection

Structured Task Knowledge | Unstructured Website Data

Question ← A wide spectrum of techniques → **Answer**

GAP

Dialog Manager (DM)

- ***A spoken conversational interface to QA***
  - *Existing literature shows promising results using text input/output, but real-time speech-to-speech systems are still in their infancy*
- ***Our Approach:***
  - *Captures generic discourse acts*
    - We trained a generic goal-oriented call classifier to classify the intent of the user into one of a predefined set of generic call types such as *vague questions, greetings, and thanks*.
    - Other requests are classified as relevant questions and directed to the QA module for further processing
  - *Selects the best answer to the user based on the QA confidence scores and dialog context*
  - *Provides a navigation mechanism when the answers are summarized in multiple segments*
  - *Interactive Question Answering*

Speech Synthesis (TTS)

- ***Challenges for Speech Synthesis in WebTalk***
  - *Awkward or unintelligible responses can dramatically reduce the perceived quality of a service.*
  - *A WebTalk system generates relatively large chunks of text blocks that are highly unsuitable for TTS.*
  - *Acronyms, abbreviations and other web specific language are highly undesirable for a TTS system.*
- ***Initial study to improve the quality of the synthesized speech***
  - *Replacing the visual structure of HTML tables, navigation bars, and other web-specific artifacts with commas, periods and TTS tags, which are used as audible cues,*
  - *Implementing changes using application-specific dictionaries*

EVALUATION

- ***Evaluate a WebTalk spoken dialog system when instantiated on a telecom company website – www.callvantage.att.com***
- ***Experimental Setup:***
  - *30 Scenarios:*
    - We manually crafted 30 scenarios, of which 24 scenarios are in-domain requests and 6 scenarios are out-of-domain. The following table provides two scenario examples. When designing these scenarios, we tried to phrase them as broad as possible so that the evaluators can express the requests in their own words.

Table 1: Scenario examples

In-Domain	You would like to know what type of hardware or equipment you would need in order to access the phone service.
Out-Domain	You are taking a trip to Florida this Thursday, and you want to check the weather out there.

Evaluation (continued)

- *- Survey*
  - We designed a web interface to present scenarios, call instructions and survey questions. Scenarios are randomly chosen. Evaluators make calls as directed by this web interface. After each call, we ask the evaluator to fill in a survey of 6 questions related to the success of the dialog. Table 2 provides the list of our survey questions

Table 2: Survey questions for the evaluation

Q1: Did you get the information you requested successfully?
Q2: When the system was unable to give you the information you wanted, were its responses sensible?
Q3: In this conversation, did the system understand what you said?
Q4: In this conversation, did you understand what the system said?
Q5: In this conversation, was it easy to find the information you wanted?
Q6: In this conversation, how would you rate your overall impression and interaction with the system?

Evaluation Results

- *- Evaluation Results*
  - In our evaluation, we received 100 calls from 16 volunteered callers. Table 3 provides a summary of the results of our experiments.

Table 3: Evaluation Results

	In-domain	Out-of-domain	Total
# dialogs	79	21	100
Q1 (% of yes)	49%	0%	37%
Q2	2.9	2.3	2.8
Q3	2.7	1.5	2.4
Q4	3.9	3.8	3.9
Q5	2.4	1.2	2.1
Q6	2.7	1.7	2.5

Evaluation Analysis

- Q!: Our results show that users were able to successfully obtain the information they requested in 49% of the dialogs for in-domain requests (see Q1). As a sanity check, this number was 0 for out-of-domain requests
- Q2: Q2 scored an average of 2.8 , which indicates the system's ability to converse with users in a sensible manner when it failed to respond with the exact answer.
- Q3: Subjects were generally not satisfied that the system "understood" them, giving Q3 an average score of 2.4 while that number is 2.7 for in-domain scenarios. This may be attributed to lower recognition and understanding accuracy.
- Q5: Q5 is related to the ease-of-use of the system and receives the lowest average of 2.1 which is certainly related to the low performance of Q3 and may also be attributed to the fact that users engaged in multiple turns before they were able to retrieve the right answer.
- Q6: In terms of overall rating (Q6), this system scored an average of 2.7 for in-domain scenarios which incidentally compares favorably with the 3.2 that was obtained for the W99 spoken dialog system. W99 was designed manually. And, models were built from a corpus of collected data.

Question Answering with Discriminative Learning Algorithms

Outline

- ***Overview Question Answering (QA)***
- **Data Driven Approaches for QA**
  - **Previous work**
  - **Our Approach**
- **Results**
  - Results on TREC data
  - Results on HandQA data
- Summary

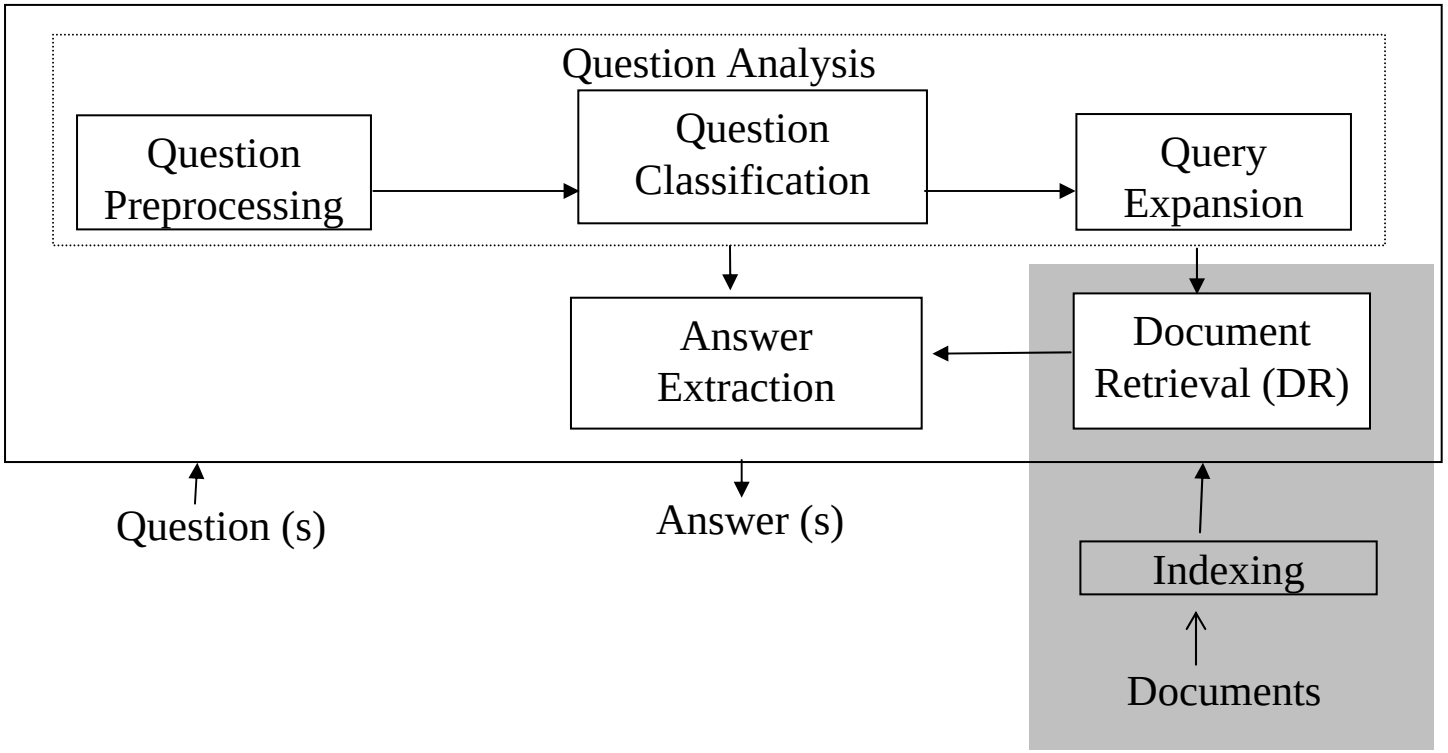
Question Answering

- ***Definition:***
  - *Question answering (QA) is an interactive human-machine process that aims to find a direct answer to a natural language question from a collection of documents.*

- ***QA systems vs. Dialog Systems***

	QA	Dialog
Domain Knowledge	Unstructured documents	Dialog Flow / Dialog plan scrip
Tasks	answer requests pertaining to the content of the given documents	Tens to Hundreds of tasks
Initiative	User initiative	*mixed-initiative*

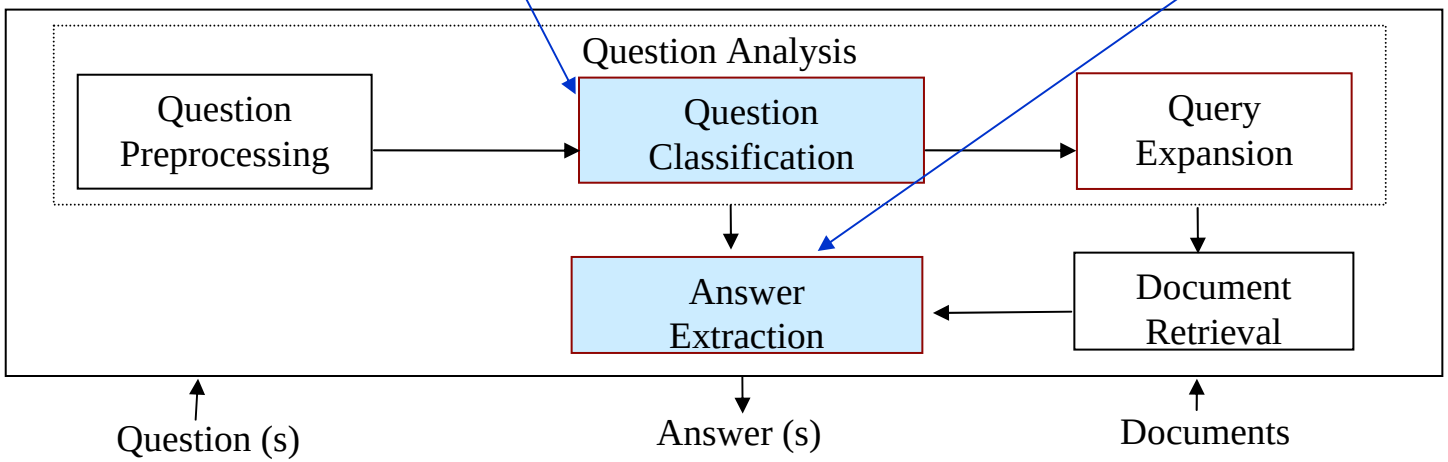
A Typical Diagram of Modern QA Systems



Data Driven Approaches for QA

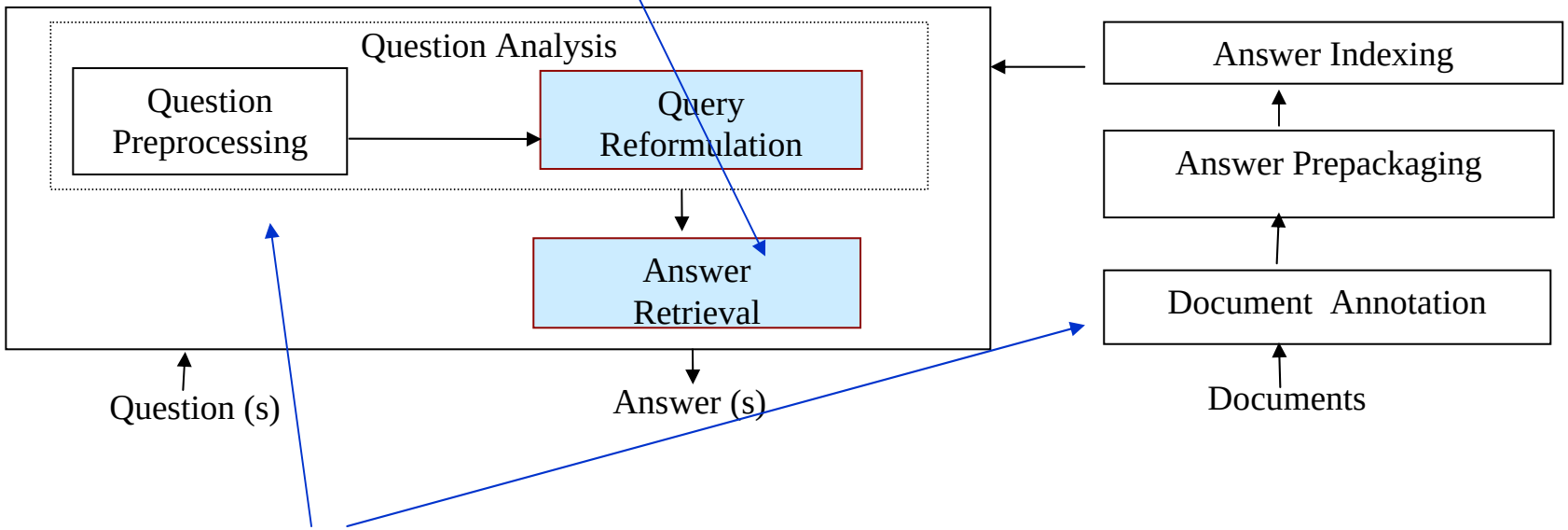
- Knowledge Intensive Approaches
- Data Driven Approaches for QA
  - Question Classification:
    - Train a **classifier** to classify questions into pre-defined classes such as "Definition", "Location-City"… [X. Li, D. Roth, "Learning Question Classifiers, 2002]
  - Answer Extraction:
    - Learn the lexical relationship between questions and answers using **statistical translation** model.

    [A.Berger etc., 2000; R. Soricut, E. Brill 2004)



Our Approach

- **A learning model** used to directly select answers from a large collection of documents



- LP technologies can contribute to the whole system.
- Training Data: 2 million FAQ-Answer pairs from the web

Answer Retrieval based on IR

- Baseline Solution: IR
  - With this framework, a natural simple solution for QA is to use the IR model.

$$score(q,a) = \sum_{w \in q,a} tf.idf(a,w) \cdot tf.idf(q,w) \cdot Z(q,a)$$

$$q = \{v_1, v_2, ... v_m\} \qquad a = \{u_1, u_2, ... u_n\}$$

Where *Z(q,a)* is a normalization factor; the *tf.idf* weights fore each word w_j appearing in a_i can be represented as:

$$tf.idf(a_i, w_j) = tf_{ij} * idf_j = tf_{ij} \cdot \log_2 \frac{N}{df_j}$$

IR is based on exact word match

The importance of each matched word is based on tf.idf

Learning needed

- *The IR model performs poorly in QA*
- *Reasons:*
  - Questions and answers are often phrased in different vocabularies and styles.
    - IR is based match is based exact word match
  - The importance of words in QA needs to be learned.
    - Within IR, *tf.idf* weights are solely determined by the answer collection. (the word *"Thailand"* is more likely to be a QA common word than the word *"take"* independent of their frequencies in the specific answer set.)
  - There is a semantic gap between questions and answers.
    - A question expresses an information need that the valid answer is expected to satisfy. (For example, a *"when"* question often expects an answer containing a *TIME /DATE* named entity value. A *"why"* question expects a reason for the matters concerned. The IR model based on exact word match doesn't provide a solution to bridge this QA semantic chasm.)

Learning Purposes

- weighting query words in terms of their importance to retrieve the correct answer;
- modeling the lexical association between questions and answers;
- modeling the semantic association between questions and answers.

Learning Algorithm: Perceptron

- We propose to use perceptron (M. Collins,2002) for this learning task
- A general linear representation:

$$score(q,a) = \phi(q,a) \cdot W$$

ϕ : A function that maps a question *q* and an answer *a* to a *K* dimensional feature vector ;

$W \in \Re^k$ A *k* dimensional parameter vector, which can be estimated through the voted percepton algorihtm

- Rewrite the IR model,

$$score(q,a) = \phi\ (q,a) \cdot W = \sum_{w \in q,a} docf\,(w,q,a) \cdot W$$

docf(w,q,a): nomalized tf.idf match score for the term *w; consistute the feature vector*

W: *W* in this case is a *K* dimension vector with all components equal to one

A Variant of the Perceptron Training Algorithm for QA

Inputs: A training set of question-answer (QA) pairs $\{(q_i, a_i)\}$ for $i = 1...n.$,where each QA pair has an associated website $site(i)$ that (q_i, a_i) appears on; A parameter *T* specifying the number of iterations over the training set; A feature function $\phi(q, a) \in \Re^k$ that maps a question-answer pair to a *k*-dimensional feature vector.

Initialization: Initialize the parameter vector *W*

Training:

For *t=1 ...T,*

Set a k-dimensional vector: $W^{sum} = 0$

For i=1 ...n

- Create a temporal parameter vector:
  $W' = W$
- Find the best answer using the current W:
  $\widetilde{a}_i = \underset{site(j)=site(i)}{\arg\max}\ \phi(q_i, a_j) \cdot W$
- If $\widetilde{a}_i \neq a_i$, then update the W':
  $W' = W' + \phi(q_i, a_i) - \phi(q_i, \widetilde{a}_i)$
- Accumulate Parameters:
  $W^{sum} = W^{sum} + W'$

Average Parameters: $W = W^{sum} / n$

Output: W

Feature Set I:

- *Exact Word Match Features*

$$score(q,a) = \phi\ (q,a) \cdot W = \sum_{w \in q,a} docf\,(w,q,a) \cdot \lambda_{tf\,.idf}\,(w)$$

$\lambda_{tf\,.idf}\,(w)$ is the parameter for characterizing the importance of the query keyword *w*.

$docf$ is the normalized *tf.idf* feature.

Feature Set-II:

- *Semantic Correlation Features*
  - Question Classification: classify a question into an NE type
  - We propose to statistically associate a question phrase qp with a Named Entity ne
    - A question phrase is defined as a stream of text at the beginning of the question
    - Example: "what scholarship" -→ "Money", "Number"
  - Learning association using **mutual information and perceptron**:
    - Find a set of *ne* for each *qp*: *NE(qp)*

$$NE(qp) = \{ne_i, I(ne_i, qp) > \sigma\}$$

$$I(qp, ne) = H(p(ne \in a)) - p(qp \in q)H(p(ne \in a \mid qp \in q))$$

    - Perceptron training: $\lambda_{sem}(qp, ne)$

$$score(q,a) = \phi(q,a) \cdot W$$

$$= \sum_{w \in q,a} docf(w,q,a) \cdot \lambda_{tf.idf}(w) + \sum_{ne \in NE(qp), qp \in q} docf(ne,q,a) \cdot \lambda_{sem}(qp,ne)$$

Feature Set III

- *Lexical Association Features*
  - Find associated answer words *{v}* for each query word *w* by calculating **mutual information**:

$$Ex(w) = \{v_i, I(v_i, w) > \sigma\}$$

  - Train the association weights $\lambda_{lex}(w,v)$ through Peceptron:

$$score(q,a) = \phi(q,a) \cdot W$$

$$= \sum_{w \in q,a} docf(w,q,a) \cdot \lambda_{tf.idf}(w) + \sum_{ne \in NE(qp), qp \in q} docf(ne,q,a) \cdot \lambda_{sem}(qp,ne)$$

$$+ \sum_{v \in Exp(w), w \in q, v \in a} docf(v,q,a) \cdot \lambda_{lex}(w,v)$$

Experiments--Data

- Training Data:
  - 2 million FAQ and answer pairs by mining the World Wide Web.
- Test :
  - TREC 2003 QA Passage Retreival data
    - Task: find an answer to a factoid question with a relatively short (250 characters) span of text
    - Answer Resource: 1,033,000 newswire documents and 3 gigabytes of text
    - Test Questions: 413 questions drawn from AOL and MSN Search logs
  - FAQ-Answer Finding:
    - Task:
    - we collected 122,363 FAQ-answer pairs as our test data, which were also mined from the Web but not included in the training data.
    - We used the 122,363 questions as the test questions and used the collection of 122,363 answers as the source from where answers would be chosen.

Experimental Results

Table 1: Experimental results: answer accuracy

Approaches		Answer Accuracy	
		TREC QA	Answer-Finding
IR		19.0%	43.0%
Perceptron Training	Feature-I	20.1%	47.5%
	Feature-I, II	28.6%	49.1%
	Feature-I, II, III	35.7%	55.5%
Absolute Improvement		16.7%	12.5%
Proximity Search		40.1%	59.%

Accuracy
0.685
0.419
0.351
0.201
0.191
0.169
0.133
0.119
0.111
0.097
0.085

Summary

- A discriminative learning approach for question answering using FAQ-Answer pairs from the web.
  - A variant of the voted perceptron training algorithm.
  - Mutual Information
- Learning include:
  - (a) weighting query words in terms of their importance to retrieve the correct answer;
  - (b) modeling the lexical association between questions and answers;
  - (c) modeling the semantic association between questions and answers.
- Test:
  - For the TREC data, we achieved 16.7% absolute improvement in answer accuracy over an IR-based baseline of 19% answer accuracy.
  - For the FAQ answer finding task, we observed 12.5% absolute improvement in answer accuracy over an IR –based baseline of 43%.

Thanks!

at&t

Your world. Delivered.
Straight to the NSA.