Question Answering and Dialogue Systems

Jan Pichl

Outline

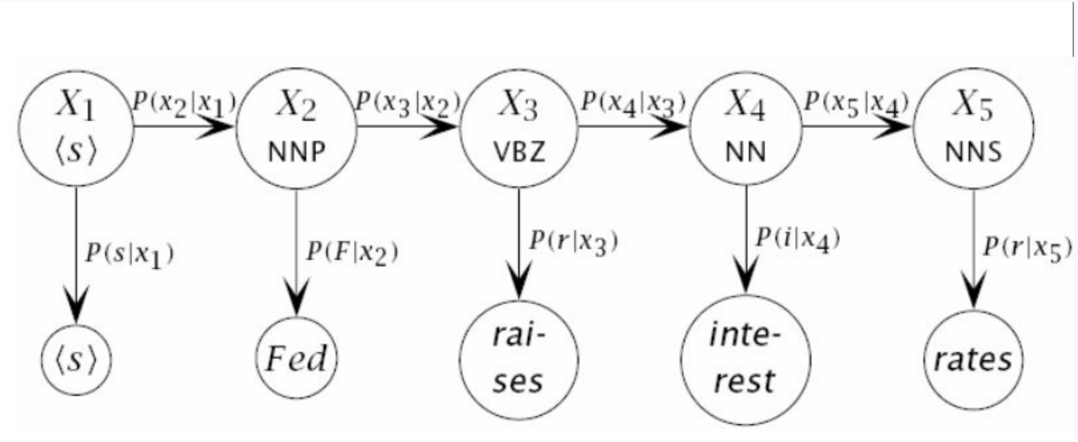
- Natural Language Processing (NLP)
  - Natural Language Understanding (NLU)
  - Natural Language Generation (NLG)
- Question Answering
  - Freetext knowledge
  - Structured knowledge
- Dialogue Systems
  - Goal oriented
  - Open domain

Part-of-speech tagging

- Hidden Markov Model
- Sequence tagging
- Nouns, Verbs, Adjectives, …
- Cca 93-95 % accuracy (English)
- Counting transition and emit counts to estimate probabilities
- Publicly available training data for many languages

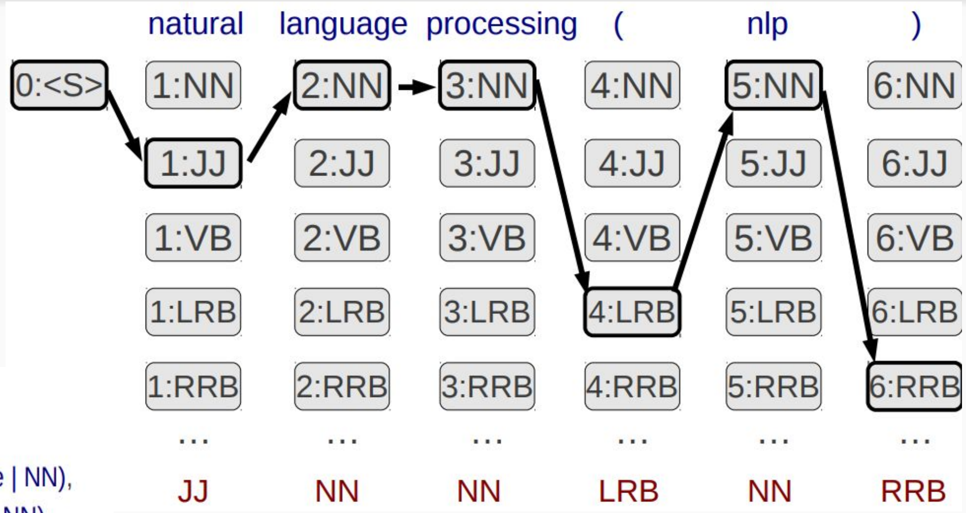
$$argmax_T \; P(T|W) = argmax_T \frac{P(W|T)P(T)}{P(W)} = argmax_T \, P(W|T)P(T)$$

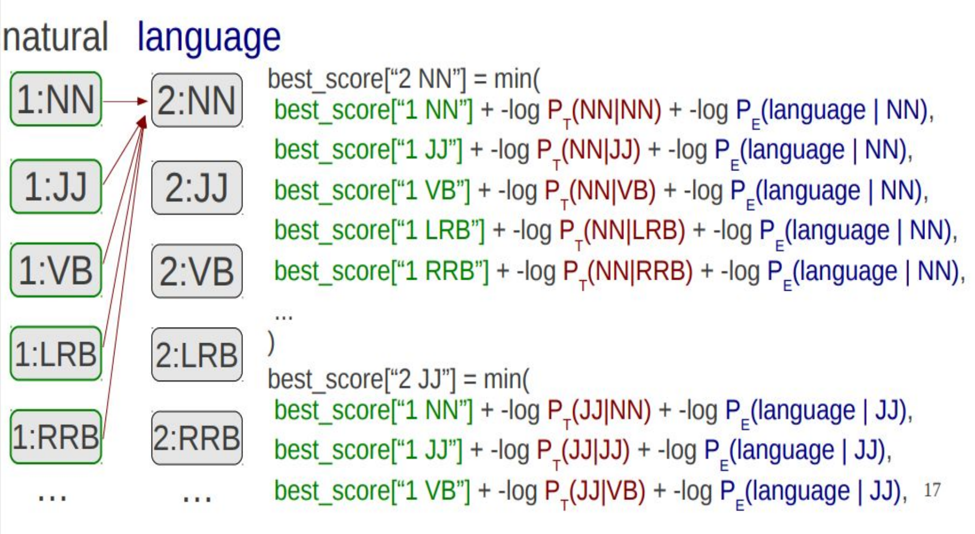
$$argmax_T \prod_{i=1}^{n} p(w_i|t_i)p(t_i|t_{i-1})$$



Part-of-speech tagging II

- Viterbi algorithm
- Dynamic programming





Question Answering

Tasks:

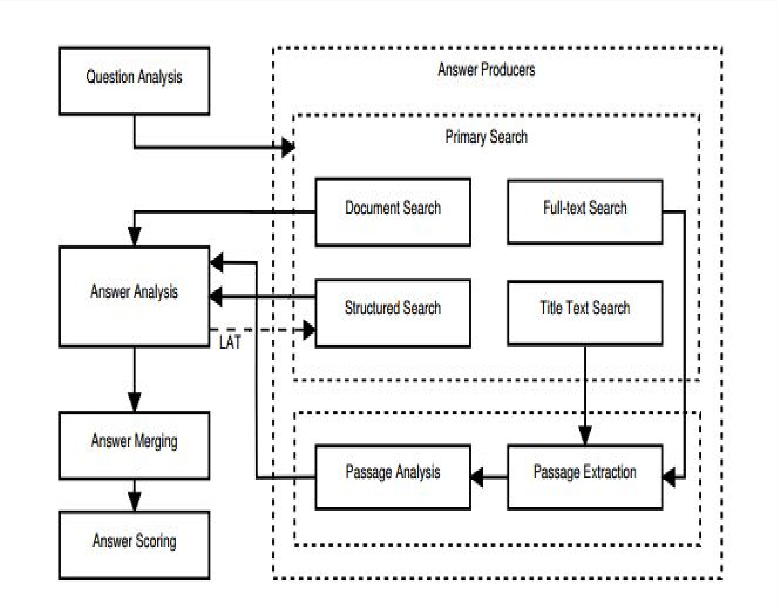
- Factoid QA
  - Most popular
  - A lot of modifications (supporting facts, list answers, yes/no answers, counting, ...)
  - IBM Watson 2011
- Visual QA
  - Questions about particular items or actions on an image
  - Combination on NLP and image processing

Approaches:

- Typically according to a knowledge source
- Information retrieval based
- Knowledge base based
- Hybrid systems - DeepQA, YodaQA

Information retrieval

- Text based method
- Takes advantage of huge amount of free text on the Web (Wikipedia, domain specific sources, ...)
- Extension of classical web search
  - Query is natural language
  - The result is an single answer which needs to be found in the search results
- Steps:
  - Question analysis
  - Answer (passage) production
  - Passage analysis
  - Answer merging and scoring



Question analysis

- POS tagging - HMM, neural network sequence tagging - Google SyntaxNex (state-of-the art)
- Entity recognition - sequence tagging HMM, CRF, usually done with linking
  - Who played meg in family guy
  - Entity: meg, family guy
- Entity linking - can be combined recognition and linking - we recognize the entity if it is successfully linked
  - Knowledge base ID
- Heuristic features:
  - Focus
    - Heuristics, based on POS and dependencies
  - Lexical answer type
    - Word from the question, describing answer, where -> location
  - Clues
    - Support verb, LAT, named entities

Answer production

- Clues in title
  - Searching for question clues in article headline
  - First sentence
- Full-text
  - Searching for clues in the whole article
  - Each sentence is considered a passage
- Concept search
  - Title and clue is an exact match
- Re-ranking of passages:
  - Features:
    - Number of named entities in passage
    - Number of question clues in passage
    - Rank of the document
    - N-gram overlap

Answer production II

- Unstructured Text Search
- Documents indexed
- Advantage of popular engines: Lucene (Solr, Elasticsearch)
- Engines based on TF-IDF and BM25
- TF-IDF:
  - Term frequency, inverse document frequency

$$\mathrm{tf_{i,j}} = \frac{n_{i,j}}{\sum_k n_{k,j}} \qquad \mathrm{idf_i} = \log \frac{|D|}{|\{j : t_i \in d_j\}|}$$

- BM25
  - Modification of TF-IDF

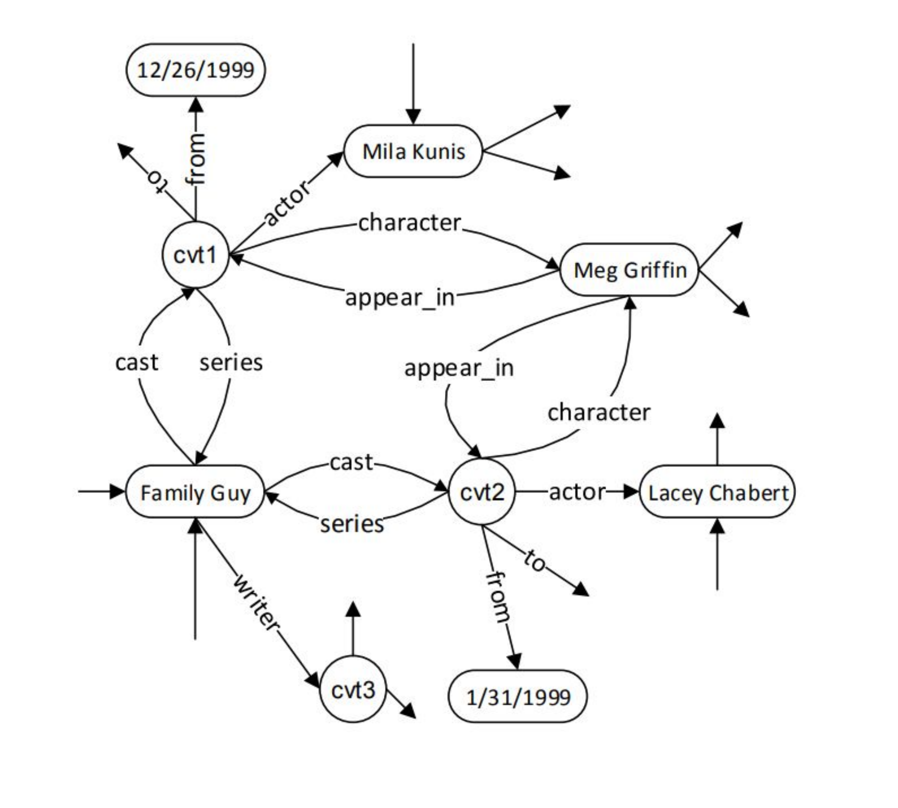
$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)},$$

Knowledge base

- Extraction of semantic representation of a query
- Mapping question representation to DB query language: SQL, SPARQL, lambda expression
- Most knowledge bases uses relations between entities - **Triple stores**
- Freebase, DBpedia, Wikidata
- Triples terminology:
    - Subject, predicate, object
    - Subject, property (relation), object
    - Entity, relation, entity

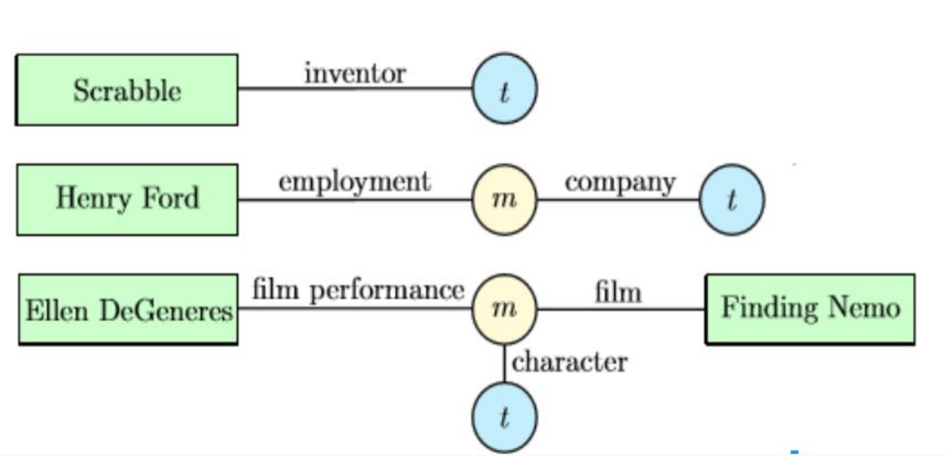
Knowledge base - structure

- Each entity (subject, object, cvt) is a graph node
- Entity - object or simple string
- CVT - compound value type, many-to-many relation
- Freebase: 44 million topics, 2.4 billion facts



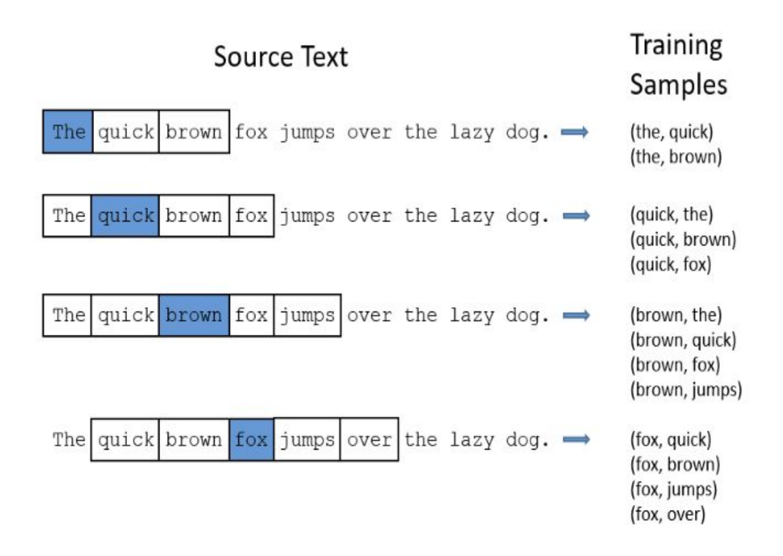
Query structure

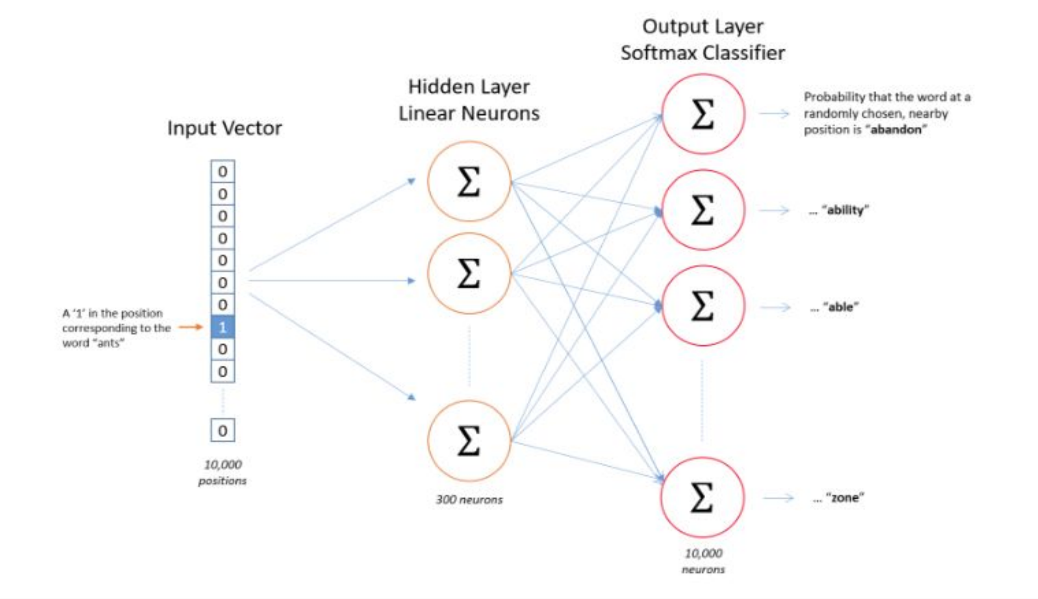
- Based on the questions from popular dataset WebQuestions (Berant et al., 2013)
- 3 basic query structures
- Sufficient for most of the questions
- Linked entity - ID of nodes in the database
- We need to find the correct relation
- Only candidates based on entity are considered



Language Modeling - word2vec

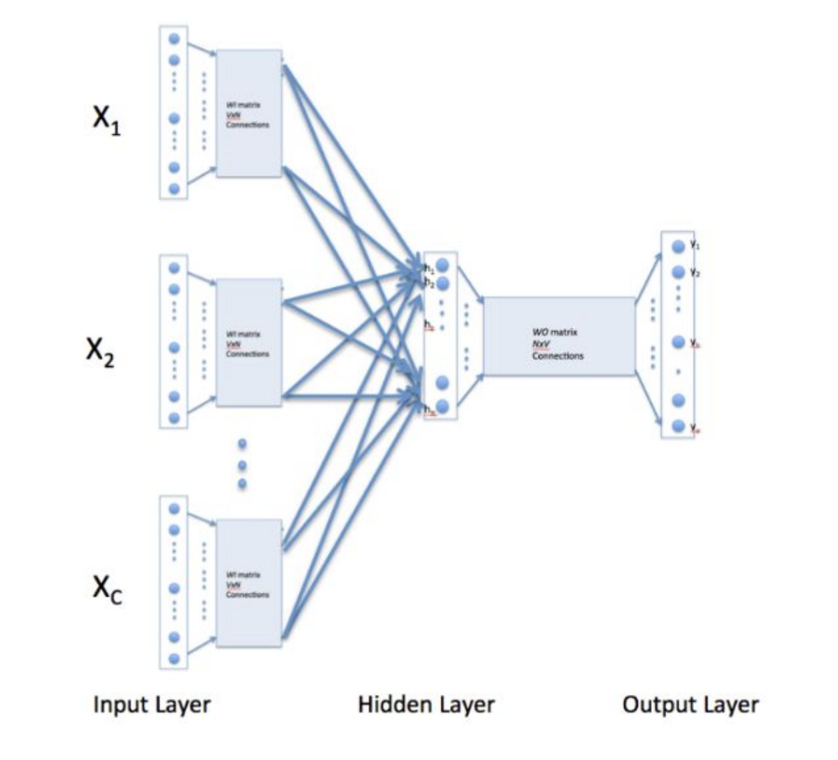
- We need to embed a word into a lower dimensional space
- Skip-gram neural network
- Arithmetic operations show some interesting relations





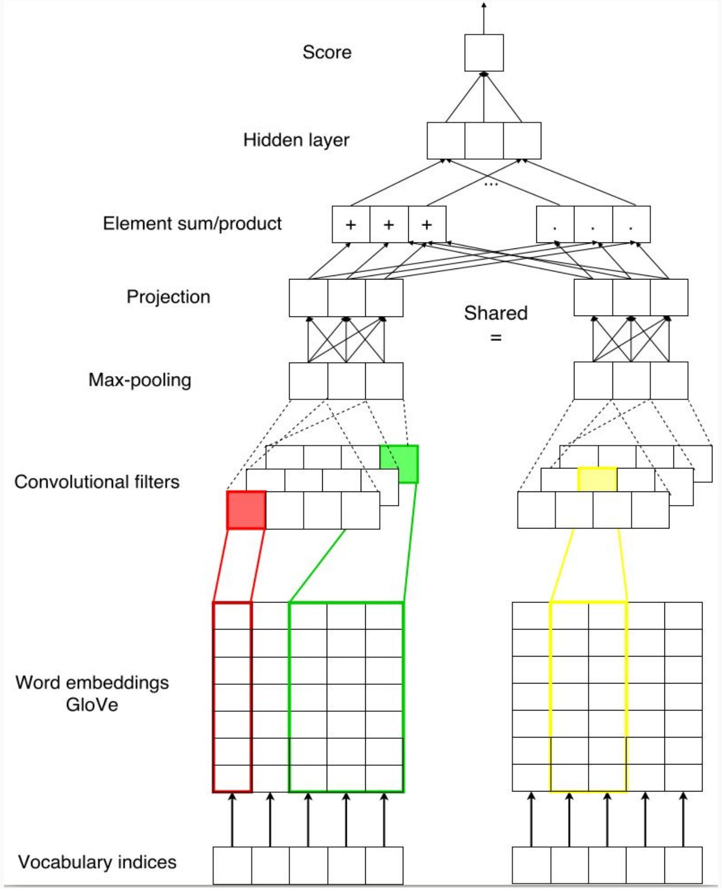
Language Modeling - word2vec II

- Continuous bag-of-words neural network



Neural network

- Similarity between question and labels of relations
- Entity masking
- Intuition: string of entity labels should have similar meaning as the question has
- Convolution filters - catches meaning of adjacent words
- Another approaches:
    - Output is actual relation instead of similarity score
    - LSTM instead of CNN



Evaluation

- Mean reciprocal rank

$$\mathrm{MRR} = \frac{1}{N} \sum_{i=1 \text{ s.t. } rank_i \neq 0}^{N} \frac{1}{rank_i}$$

- Accuracy at one
  - Percentage of questions with correct answer with the highest score
- AP Recall
  - Number of questions with correct answer even considered (regardless the rank)

Dialogue Systems

- Historical systems:
    - ELIZA
    - ALICE
- Goal oriented
    - Conversation ends when the goal is reached
    - Shorter conversation is usually better
    - Hotel/Restaurant reservation
- Chit-chat/Open domain:
    - Without specific goal
    - The goal is to entertain an user as long as possible
    - User decides when is the end of the conversation

ELIZE

- Joseph Weizenbaum, 1964 - 1966
- Scenario: the therapist
- Keyword detection
- Handcrafted responses
- Using parts of the user message as a response

> Hello, I am Eliza.
* Hello
> How are you today.. What would you like to discuss?
* I am depressed.
> Did you come to me because you are depressed?
* Yes
> I see.
* It is because I have problem with my parents.
> Can you elaborate on that?
* Yes, I think it is because of my dad.
> Do you really think so?
* Yes, he can be really moody sometimes.
> I understand.

ALICE

- Artificial Linguistic Internet Computer Entity
- Richard Wallace, 1995
- Inspired by ELIZA
- Custom language: AIML - Artificial intelligence markup language

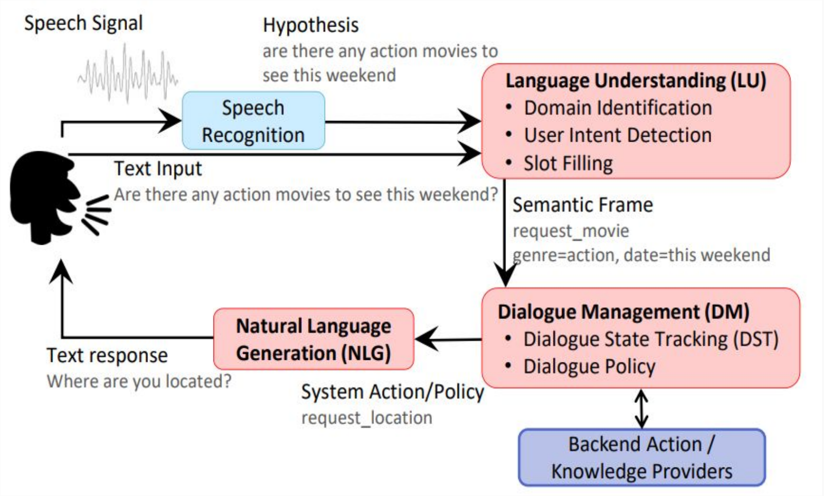
```
<?xml version = "1.0" encoding = "UTF-8"?>
<aiml version = "1.0.1" encoding =
"UTF-8"?>
   <category>
      <pattern>I am *</pattern>
      <template>
         Hello <set name = "username">
<star/>! </set>
      </template>
   </category>

   <category>
      <pattern>Good Night</pattern>
      <template>
         Hi <get name = "username"/>
Thanks for the conversation!
      </template>
   </category>
</aiml>
```

Goal oriented dialogues

- Combination of rules and statistical components
  - POMDP for spoken dialog systems (Williams and Young, 2007)
  - End-to-end trainable task-oriented dialogue system (Wen et al., 2016)
  - End-to-End Task-Completion Neural Dialogue Systems (Li et al., 2017)

Dialogue components

- Typical structure of goal oriented dialogue
- Speech recognition hypotheses
- Intent (find_restaurant, find_movie, give_information)
- Slot-value pairs (food_type=asian)
- Knowledge retrieval
- Dialogue management
- Natural language generation

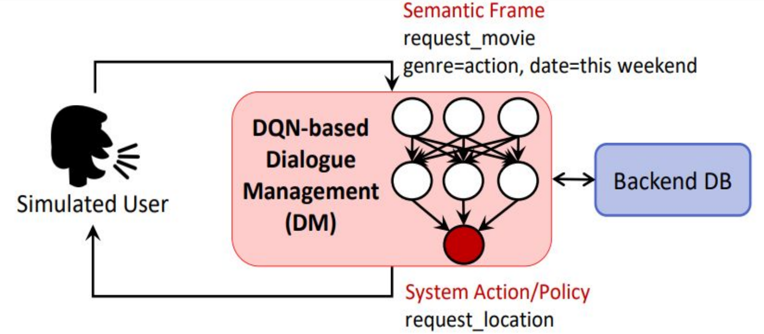


Intent detection and slot filling

- Can be divided into two separate tasks or processed simultaneously
- Intent detection
  - Classification of the input sentence into a intent class
- Slot filling
  - Sentence labeling
  - Classes: Outside, Begin-slot_type, Inside-slot_type
  - HMM, CRF, LSTM networks
- Combined solution:
  - LSTM network, last output is the intent
  - Input: w_1, w_2, … ,w_n, <eos>
  - Output: y_1, y_2, … , y_n, i

Dialogue state tracking

- A.k.a. Dialogue management (DM)
- Input: intent and slot-value pairs
- Forming database query
- Deep Q Network, input: current state, output: action
- ϵ-greedy exploration
- Experience replay
- Issues: cold start, slow learning

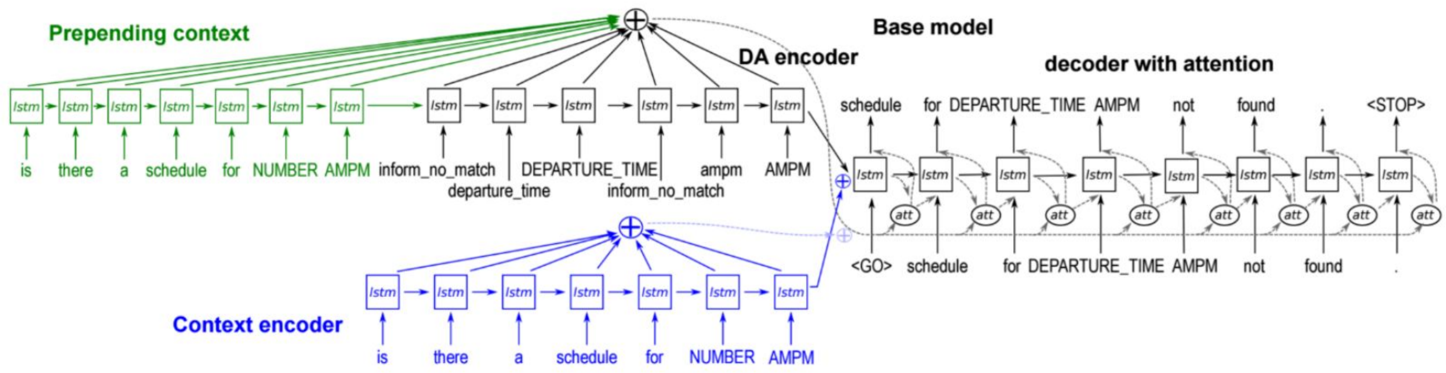



```
for e in range(EPISODES):
    state = sim.reset()
    for time in range(500):
        action = agent.act(state)
        next_state, reward, done = sim.step(action)
        agent.remember(state, action, reward, next_state, done)
        state = next_state
        if done:
            break
    if len(agent.memory) > batch_size:
        agent.replay(batch_size)
```

```
def replay(self, batch_size):
    minibatch = random.sample(self.memory, batch_size)
    for state, action, reward, next_state, done in minibatch:
        target = reward
        if not done:
            target = reward + self.gamma *
np.amax(self.model.predict(next_state)[0])
        target_f = self.model.predict(state)
        target_f[0][action] = target
        self.model.fit(state, target_f, epochs=1, verbose=0)
    if self.epsilon > self.epsilon_min:
        self.epsilon *= self.epsilon_decay
```

Natural language generation

- Simplest method - template based NLG
  - confirm(food=$V) “Do you want a $V restaurant?”
- Pros: simple, error-free, easy to control
- Cons: time-consuming, poor scalability
- Sequence-to-sequence network
- Input is the sequence of triples intent-slot-value
- Output is a natural language sentence

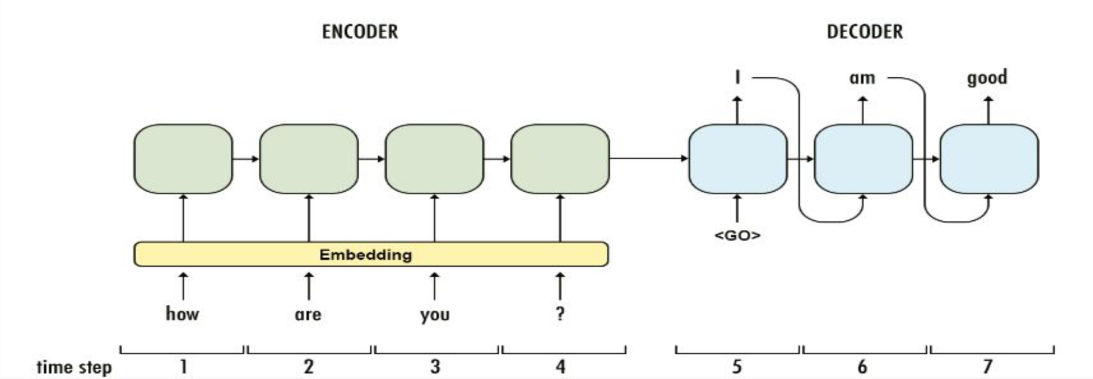


Open domain dialogues

- Cannot be distinguished between successful and unsuccessful dialogue
- Using variants of seq2seq model
  - Inspired by machine translation
- A neural conversation model (Vinyals and Le, 2015)
- Reinforcement learning for dialogue generation (Li et al., 2016)

Sequence to sequence

- Mapping input sentence to response sentence
- Encoder - decoder
- Single input sentence or multiple dialogue turns to preserve the context
- Problems
  - Objective function does not capture the goal of the dialogue (longer responses instead of single words, informative responses instead of generic “I’dont know”
  - Large and good quality data set of human conversations



Reinforcement learning for dialogue generation

- Modification of the seq2seq approach
- Addresses the issues with non-informative and generic responses
- Supervised training of seq2seq - it is used to compute rewards for reinforcement learning
- Rewards:
    - Ease of answering
        - List of dull responses
        - Negative log prob of dull response given action (based on pre-trained model)
    - Information flow
        - Penalizing semantic similarity between two consecutive answers of the same agent
        - Negative log cosine similarity
    - Semantic coherence
        - Probability of generating response a given the previous dialogue utterances plus
        - Backward probability of generating the previous dialogue utterance based on the response

Handcrafted dialogue structure with trained management

- Motivation:
  - The responses needs to be precisely prepared by dialogue maker
  - More engaging responses
  - Avoiding profanity
- Graph structure of dialogue
- Top-level dialogue management (DM)
  - Selects a suitable dialogue graph
  - Classification of the sentence
- Topic-level DM
  - Navigates in the graph structure
  - Classification of the sentence
  - Selects a graph node

Thank you!

Bibliography

[1] JURAFSKY, Dan; MARTIN, James H. *Speech and language processing*. London: Pearson, 2014.

[2] BAUDIŠ, Petr. YodaQA: a modular question answering system pipeline. In: *POSTER 2015-19th International Student Conference on Electrical Engineering*. 2015. p. 1156-1165.

[3] YIH, Scott Wen-tau, et al. Semantic parsing via staged query graph generation: Question answering with knowledge base. 2015.

[4] Berant, A. Chou, R. Frostig, and P. Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In Empirical Methods in Natural Language Processing (EMNLP)

[5] Chen, Y. Celikyilmaz, A. Hakkani-Tur D. 2017. Tutorial - Deep Learning for Dialogue Systems