Question Answering

Ramaseshan Ramachandran

QUESTION ANSWERING - INTRODUCTION

An application of short Post-Response is Question Answering system, such as IBM Watson (Jeopardy). In this case most of the candidate responses are answers for factoid questions

- Open domain question answering has become important research area in natural language processing
- Tougher than common search engine tasks
  - Finding accurate and concise answers to questions rather than a set of relevant document
- Simple term-based retrieval won't be enough
- **Type** of the sought after answer should be known to retrieve accurate answers

QUESTION ANSWERING - SAMPLES

Question	Hierarchy	Type
What is RNN?	Abbreviation	Expansion
Where is the big temple in India located?	Location	City
Who was the president of India in 2006?	Human	Person
Name the currency used in China	Entity	Currency
How far away is the moon?	Numeric	Distance
What is the chemical symbol for oxygen?	Entity	Symbol
What is a prism?	Description	Definition
Why is the sun yellow?	Description	Reason
When did CV Raman receive his Nobel Prize?	Numeric	Year

Most questions could be classified in to 6 major classes[1] - ABBREVIATION, ENTITY,DESCRIPTION, HUMAN, LOCATION and NUMERIC VALUE and around 50 fine-grained types.

[1] Xin Li, Dan Roth, Learning Question Classifiers

DEFINITION OF QUESTION CLASSES

Class	**Definition**
ABBREVIATION	abbreviation
abb	abbreviation
exp	expression abbreviated
ENTITY	entities
animal	animals
body	organs of body
color	colors
currency	currency names
dis.med.	diseases and medicine
LOCATION	locations
city	cities
country	countries
mountain	mountains
...	...
NUMERIC	numeric values
code	postcodes or other codes
date	dates
DESCRIPTION	description and abstract concepts
definition	definition of sth.
...	...
HUMAN	human beings
group	a group or organization of persons
ind	an individual
title	title of a person
description	description of a person
...	...

Reference: Definition of Question Classes

EXPERIMENTAL DATA FOR QUESTION CLASSIFICATION

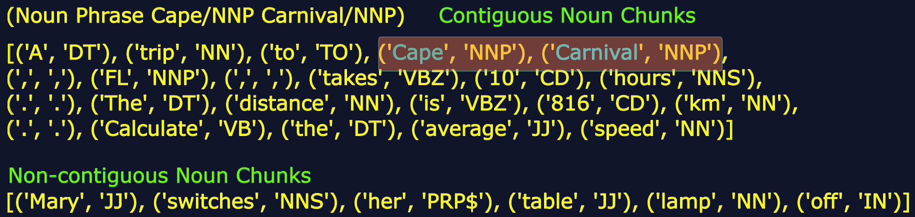
```
ENTY:cremat What films featured the character Popeye Doyle ?
ENTY:animal What fowl grabs the spotlight after
            the Chinese Year of the Monkey ?
ABBR:exp What is the full form of .com ?
LOC:city What city did the Flintstones live in ?
LOC:other Where is the Kalahari desert ?
HUM:gr What company tabulates the ballots in
           voting for the Academy Awards ?
DESC:desc What do Mormons believe ?
NUM:money How much money does a back injury lawsuit get ?
NUM:other What was Einstein's IQ ?
NUM:count How many people died on D-Day ?
ABBR:exp What does 'PSI' stand for ?
...
```

Reference: Training Data for Text Retrieval Conference (TREC)

FEATURE SPACE

- Words
- Part of Speech (POS) tags
- Chunks(non-overlapping phrases)
- Named entities
- Head chunks(using POS - first noun chunk[2] in a sentence)[3]
- Semantically related words (words that often occur with a specific question class - How far, How high, How long)

Contiguous chinking - Example



(Noun Phrase Cape/NNP Carnival/NNP)

Contiguous Noun Chunks

```
[('A', 'DT'), ('trip', 'NN'), ('to', 'TO'), ('Cape', 'NNP'), ('Carnival', 'NNP'),
(',', ','), ('FL', 'NNP'), (',', ','), ('takes', 'VBZ'), ('10', 'CD'), ('hours', 'NNS'),
('.', '.'), ('The', 'DT'), ('distance', 'NN'), ('is', 'VBZ'), ('816', 'CD'), ('km', 'NN'),
('.', '.'), ('Calculate', 'VB'), ('the', 'DT'), ('average', 'JJ'), ('speed', 'NN')]
```

Non-contiguous Noun Chunks

```
[('Mary', 'JJ'), ('switches', 'NNS'), ('her', 'PRP$'), ('table', 'JJ'), ('lamp', 'NN'), ('off', 'IN')]
```

Non-contiguous phrase example Mary switches her table lamp off

[2] In English grammar, a head is the key word that determines the nature of a phrase

CHUNKING USING NLTK

```
import nltk
from nltk.tokenize import word_tokenize
from nltk import word_tokenize, pos_tag

def regex_chunker(text, regex):

    sent = pos_tag(word_tokenize(text))
    cp = nltk.RegexpParser(regex)
    chunked_pos_tags = cp.parse(sent)

    for chunk in chunked_pos_tags.subtrees(
                    filter=lambda t: t.label() == 'Noun Phrase'):
        print(chunk)

if __name__ == '__main__':
    text = '''A trip to Cape Carnival, FL, takes 10 hours.
                    The distance is 816 km. Calculate the average
   speed'''
    regex_chunker(text, r'Noun Phrase: {<NN.?>+<NN.?>}')
```

QUESTION TYPOLOGY RULES

Simple rules could be defined to classify questions

For example,

```
1. if QuestionStartsWith(who) or QuestionStartsWith(whom)
       TopHierarchy ← HUMAN
       Class ← PERSON
   fi
2. if QuestionStartsWith(where)
       TopHierarchy ← LOCATION
       Class ← CITY
   fi
```

If a query contains Which or What, then the head noun phrase determines the class, as for What X questions

What is a prism?

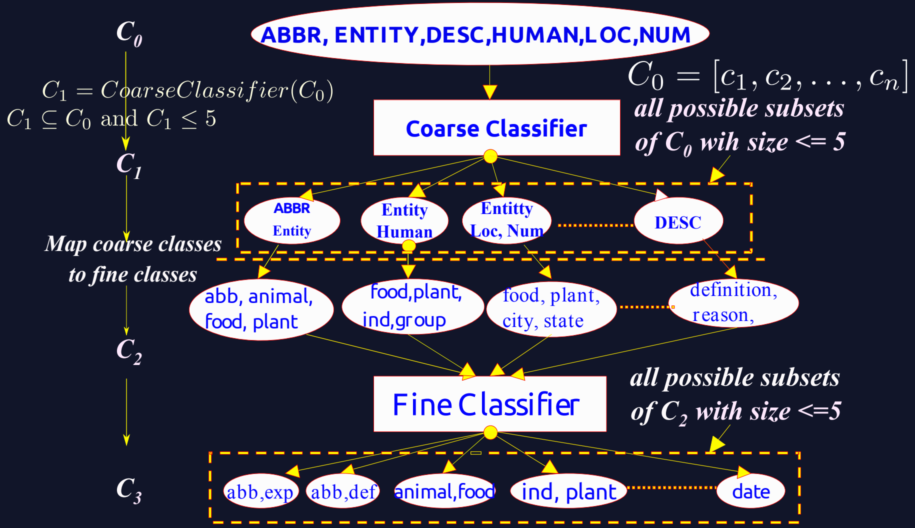
LEARNING QUESTION CLASSIFIERS

- Answering factual questions[1]
- understanding the question's intent and the type of answer it seeks
- Classifying questions into categories
- Constraining the search for potential answers
- Guide answer verification

- Uses lexical features for classification

  Lexical features include words, phrases, and part-of-speech tags extracted from the question
- The authors use a two-layer question taxonomy with coarse-grained (e.g., definition, location) and fine-grained (e.g., person, city) categories
- Different machine learning algorithms (Naive-Bayes)can be used using the labeled question-answer pairs

LEARNING QUESTION CLASSIFIERS



C_0

ABBR, ENTITY,DESC,HUMAN,LOC,NUM

$$C_1 = CoarseClassifier(C_0)$$

$$C_1 \subseteq C_0 \text{ and } C_1 \leq 5$$

$$C_0 = [c_1, c_2, \ldots, c_n]$$

all possible subsets of C_0 wih size <= 5

Coarse Classifier

C_1

ABBR Entity

Entity Human

Entitty Loc, Num

DESC

Map coarse classes to fine classes

abb, animal, food, plant

food,plant, ind,group

food, plant, city, state

definition, reason,

C_2

Fine Classifier

all possible subsets of C_2 with size <=5

C_3

abb,exp

abb,def

animal,food

ind, plant

date

DECISION RULE

Given the list of classes and the features for each of the question, it is easy to calculate the probability distribution of classes for the given question[1]
The probability density is

$$P = [p_1, p_2, \ldots, p_n] \tag{1}$$

and the corresponding class labels are

$$C = [c_1, c_2, \ldots, c_n] \tag{2}$$

p_is are obtained by employing Naive-Bayes algorithm

QA Using Neural Models

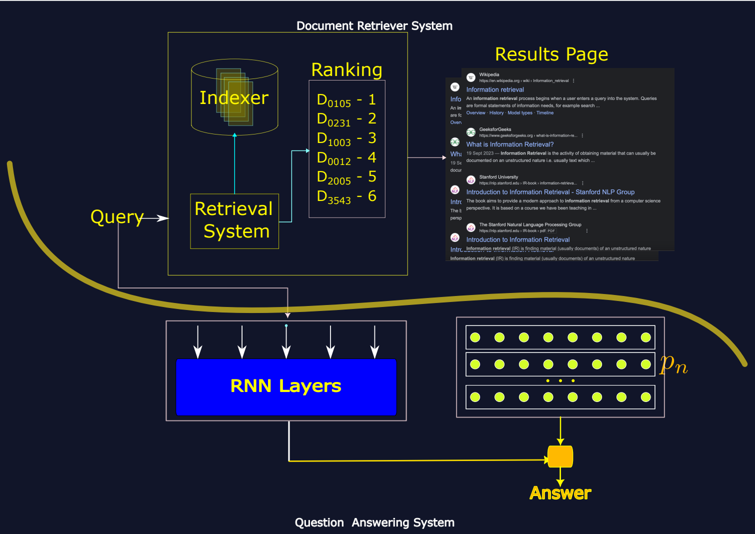
READING COMPREHENSION

Reading comprehension task seeks to estimate the conditional probability

$$p(a \mid c, q) \tag{3}$$

where c is a context document
a query relating to that document, q and
the answer to that query, a.

COMPONENTS OF A QA SYSTEM



Wikipedia
https://en.wikipedia.org › wiki › Information_retrieval

Information retrieval

An **information retrieval** process begins when a user enters a query into the system. Queries are formal statements of information needs, for example search ...

Overview · History · Model types · Timeline

GeeksforGeeks
https://www.geeksforgeeks.org › what-is-information-re...

What is Information Retrieval?

19 Sept 2023 — **Information Retrieval** is the activity of obtaining material that can usually be documented on an unstructured nature i.e. usually text which ...

Stanford University
https://nlp.stanford.edu › IR-book › information-retrieva...

Introduction to Information Retrieval - Stanford NLP Group

The book aims to provide a modern approach to **information retrieval** from a computer science perspective. It is based on a course we have been teaching in ...

The Stanford Natural Language Processing Group
https://nlp.stanford.edu › IR-book › pdf PDF

Introduction to Information Retrieval

Information retrieval (IR) is finding material (usually documents) of an unstructured nature

Information retrieval (IR) is finding material (usually documents) of an unstructured nature

Figure: QA System

IDENTIFYING SPAN OF TOKENS

Who is CV Raman?

Sir CV Raman (7 November 1888-21 November 1970) was an Indian physicist born in the former Madras Province in India (presently the state of Tamil Nadu), who carried out ground-breaking work in the field of light scattering, which earned him the 1930 Nobel Prize for Physics. He discovered that when light traverses a transparent material, some of the deflected light changes wavelength and amplitude. This phenomenon, subsequently known as Raman scattering, results from the Raman effect[4] In 1954, the Indian government honored him with India's highest civilian award, the Bharat Ratna [5][6]

IDENTIFYING SPAN OF TOKENS

What is the invention of CV Raman?

Sir CV Raman (7 November 1888-21 November 1970) was an Indian physicist born in the former Madras Province in India (presently the state of Tamil Nadu), who carried out ground-breaking work in the field of light scattering, which earned him the 1930 Nobel Prize for Physics.

He discovered that when light traverses a transparent material, some of the deflected light changes wavelength and amplitude.

This phenomenon, subsequently known as Raman scattering, results from the Raman effect[4] In 1954, the Indian government honored him with India's highest civilian award, the Bharat Ratna [5][6]

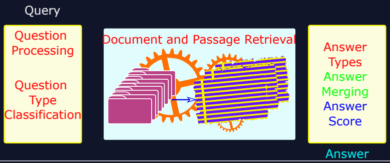
ANSWER EXTRACTION

The important phase ⌣ in the QA system

Span Labeling: The span of text (tokens) that contains the answer. The task of finding the span of text is known as Span Labeling

Modern approaches combine a IR-based component based on bigram hashing and TF*IDF matching and a multi-layer recurrent neural network model trained to detect answers [2] [3]

Emerging systems are designed as reading comprehension systems



Two important components

- **Document Retriever**: Utilizes bigram hashing and TF-IDF matching to efficiently return relevant articles based on a given question.
- **Document Reader**: A multi-layer recurrent neural network model trained to extract answer spans from the retrieved documents.

DOCUMENT RETRIEVER

- Using a typical Term-Document and the retrieval operations on the Term-Document matrix
- Using Inverted Indexing approach used in SOLR/Elastic search
- Using LSA
- Combination of the above with n-grams
- Using a ranking model to retrieve top 5-10 documents
- Use an answer encoder to find similar representations in the documents - Use of RNN

FEATURES FOR ANSWERS

- Phrase matches keywords/patterns of question and the paragraph
- Count of terms that match question and potential paragraphs
- Cosine similarity
- Pattern matching using trained ANNs
- Probabilistic methods using alignment methods

QUESTION ENCODING

- A question encoder creates weighted sum of all the words (q_i) in a question.
- The word embedding of each word in the question is fed to an RNN encoder
- For every time state, q_i, a hidden $\mathbf{q_i}$ is output from the hidden unit.
- For all the time states, a weighted sum $\mathbf{q}$ and a single embedding of the question is the output - $\mathbf{q} = [\mathbf{q_1}, \mathbf{q_2}, \mathbf{q_3}, \ldots \mathbf{q_l}]$

$$\mathbf{q} = \sum_j b_j q_j \tag{4}$$

$$b_j = \frac{\exp(\mathbf{w.q}_j)}{\sum_i^t \exp(\mathbf{w.q}_i)} \tag{5}$$

where $\mathbf{w}$ is the weight vector to be learned

PARAGRAPH ENCODING

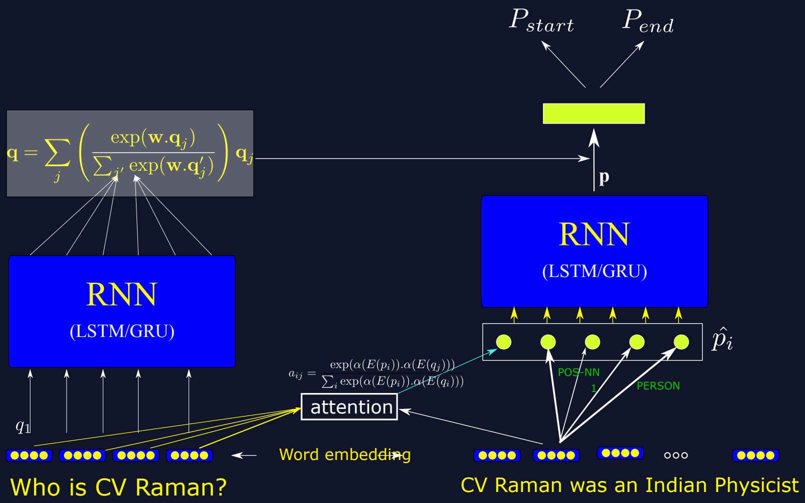
Let $q = (q_1, q_2, \ldots, q_n)$ be the question with n tokens
Let $\mathbf{p} = (\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_m})$ be the encoded paragraphs of $\hat{p} = (\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_m)$
and $\hat{p}_i$ represent the following:

1. The embedding of the word $f_1 = E(p_i)$
2. p_i can be matched exactly by one question word $f_2 = \mathbb{1}(p_i \in q_i)$
3. Token feature such as POS, NER, TF/TF*IDF - $f_{features}$
4. Aligned question embedding $f_{align}(p_i) = \sum_j a_{ij} E(q_j)$, where a_{ij} captures the similarity between p_i and q_j

$$a_{ij} = \frac{\exp(\alpha(E(p_i)).\alpha(E(q_j)))}{\sum_{j'} \exp(\alpha(E(p_i)).\alpha(E(q_j')))} \tag{6}$$

$\alpha(.)$ is a single dense layer with ReLU nonlinearity. Compared to the exact match features, these features add soft alignments between similar but non-identical words(e.g.,car and vehicle)

QA SYSTEM



$$\mathbf{q} = \sum_j \left(\frac{\exp(\mathbf{w}.\mathbf{q}_j)}{\sum_{j'} \exp(\mathbf{w}.\mathbf{q}'_j)} \right) \mathbf{q}_j$$

$$a_{ij} = \frac{\exp(\alpha(E(p_i)).\alpha(E(q_j)))}{\sum_i \exp(\alpha(E(p_i)).\alpha(E(q_i)))}$$

PREDICTION

- The goal is to predict the span of tokens that is most likely the correct answer
- The RNN is trained using paragraph vectors $(\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m)$ and question vector $\mathbf{q}$ to predict the span (P_{start}, P_{end})
- A bilinear attention layer **W** is used to predict instead of a simple similarity measure as follows:

$$P_{start_i} \propto \exp(p_i \mathbf{W} \mathbf{q}) \tag{7}$$

$$P_{end_i} \propto \exp(p_i \mathbf{W} \mathbf{q}) \tag{8}$$

- During prediction, the best span from $token_i$ to $token_{i'}$ such that $i \leq i' \leq i + 15$ and $P_{start}(i) \times P_{end}(i')$ is maximized.
- Answer $= \arg\max_j (P_{start}(i) \times P_{end}(i'), j = 1 \cdots, n)$

EXPERIMENTS

- 3-layer bidirectional LSTMs with $h = 128$ hidden units for both paragraph and question encoding
- Stanford CoreNLP toolkit for tokenization and also generating lemma, part-of-speech, and named entity tags

Features	F1
Full	78.8
No f_{token}	78.0 (-0.8)
No f_{exact_match}	77.3 (-1.5)
No $f_{aligned}$	77.3 (-1.5)
No $f_{aligned}$ and f_{exact_match}	59.4 (-19.4)

EVALUATION OF THE CONVERSATION AGENTS

Most of the researchers use F1 score It is a weighted harmonic mean of *Precision* and *Recall* given by the relation:

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P + R}, where, \beta^2 = \frac{1-\alpha}{\alpha} \tag{9}$$

where $\alpha \in \{0,1\}$ and $\beta \in \{0,\infty\}$. When $\alpha = \frac{1}{2}$ or $\beta = 1$, it is a balanced measure that gives equal weights to *Precision* and *Recall*

$$F_{\beta=1} = F_1 = \frac{2PR}{P+R} \tag{10}$$

$$Precision = \frac{\# of relevant items}{\# of retrieved items} \tag{11}$$

$$Recall = \frac{\# of relevant items retrieved}{\#, of Relevant items} \tag{12}$$

	Relevant	Not relevant
Retrieved	TP	FP
Not Retrieved	FN	TN

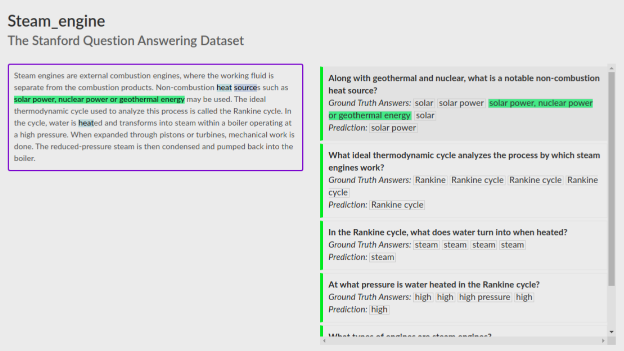
$$Precision = TP/(TP+FP) \tag{13}$$

$$Recall = TP/(TP+FN) \tag{14}$$

DATA SETS FOR READING COMPREHENSION TRAINING

Stanford Question Answering Dataset (SQuAD)

- Reading Comprehension Data set
- 87000 examples for training and 10000 examples for development
- All questions and answers are composed by humans through crowd sourcing
- The span of text is provided for all questions that could be answered



Steam_engine

The Stanford Question Answering Dataset

Steam engines are external combustion engines, where the working fluid is separate from the combustion products. Non-combustion heat sources such as solar power, nuclear power or geothermal energy may be used. The ideal thermodynamic cycle used to analyze this process is called the Rankine cycle. In the cycle, water is heated and transforms into steam within a boiler operating at a high pressure. When expanded through pistons or turbines, mechanical work is done. The reduced-pressure steam is then condensed and pumped back into the boiler.

Along with geothermal and nuclear, what is a notable non-combustion heat source?
Ground Truth Answers: solar | solar power | solar power, nuclear power or geothermal energy | solar
Prediction: solar power

What ideal thermodynamic cycle analyzes the process by which steam engines work?
Ground Truth Answers: Rankine | Rankine cycle | Rankine cycle | Rankine cycle
Prediction: Rankine cycle

In the Rankine cycle, what does water turn into when heated?
Ground Truth Answers: steam | steam | steam | steam
Prediction: steam

At what pressure is water heated in the Rankine cycle?
Ground Truth Answers: high | high | high pressure | high
Prediction: high

What types of engines are steam engines?

Datasets used: Stanford Question Answering Dataset-SQuAD,CuratedTREC,

WebQuestions and WikiMovies [4]

BIDIRECTIONAL ATTENTION FLOW MODEL

- Utilizes a multi-stage hierarchical process to represent context[5]
- Concatenates character and word embedding vectors
- Contextual embedding $H \in \mathbb{R}^{\nvDash \times \mathbb{T}}$ is obtained from context word vectors, X
- Contextual Query vector $U \in \mathbb{R}^{\nvDash \times \mathbb{J}}$ is obtained from query word vectors Q.
- Couples query and context vectors in two directions - context to query and from query to context.
- Context-to-query (C2Q) attention - which query words are most relevant to each context word
- Query-to-context (Q2C) attention- context words have the closest similarity to one of the query words
- The above steps encode the query-aware representations of context words
- Modeling Layer: Employs bi-directional LSTM to capture interactions among context words conditioned on the query.
- Output Layer: Predicts answers by finding sub-phrases in the paragraph based on start and end indices

BIDAF

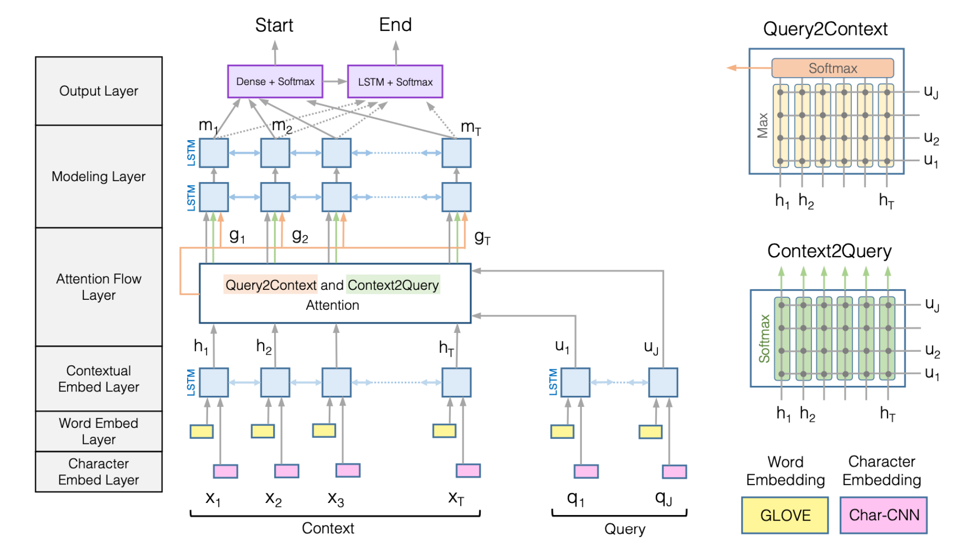


Figure: Bidirectional Attention Flow Model

SPEECH ACT

We could also use Speech Act to make the conversation more robust and human-like by using them as features.

Each utterance in a dialog is closely related to an action. Speech acts refer to the various functions that language can perform beyond conveying information

- **Assertives**: Statements that convey information.
- **Directives**: Commands or requests.
- **Commissives**: Promises, commitments.
- **Expressives**: Expressions of emotion or attitude.
- **Declarations**: Changes the state of affairs
- **Acknowledgements**

Thus speech act expresses an important component of the intention of the spoken/written sentences

LOCUTIONARY ACT

This basic level of communication focuses on the literal meaning conveyed by the words themselves

Examples

- Coveys directly the literal meaning
- Doesn't involve intention, context, or interpretation
- Essential for building more complex speech acts.
    - `The sky is blue` - simple statement
    - `What time is it?` - question $\smile$
    - `Stop` - commands
    - `Congratulations`- Exclamations
    - `I promise`- Performatives

ILLOCUTIONARY ACT

Each of these sentences not only conveys information but also performs a particular illocutionary act, such as expressing a belief, making a request, committing to an action, expressing emotions, or making a declaration.

Examples

- `I believe that the sky is blue` - assertive
- `Please pass the salt` - Directive
- `I promise to meet you at 5 pm` - Commissive
- `Congratulations on your promotion!` - Expressive
- `I declare this meeting adjourned` - Declarative

SPEECH ACT WITH EXAMPLES

Sentence type	Examples	Force
Declarative	Turtles are amazing.	assertion
	I wonder where Kim is.	question
	You should move your bicycle.	suggestion
	You can have a cookie.	invitation
	It would be a shame if something happened to your store.	threat
Interrogative	Is today Tuesday?	question
	What day is today?	question
	What on earth are you doing?	accusation
	Do you want to have ice-cream?	invitation
	Could you help me?	request
Imperative	Move your bicycle!	command
	Have a cookie.	invitation
	Please rain!	plea
	Get well soon!	well-wish
	Turn right here.	request

Source: Speech Act

GROUNDING

Any dialog is not an independent set of sentences. Dialogs happen on a **common ground**. The speakers often perform ground Grounding act by acknowledging that the hearer has understood the speaker

REFERENCES

[1] Xin Li and Dan Roth. "Learning Question Classifiers". In: *COLING 2002: The 19th International Conference on Computational Linguistics*. 2002. URL: https://aclanthology.org/C02-1150.

[2] Karl Moritz Hermann et al. *Teaching Machines to Read and Comprehend*. 2015. arXiv: https://arxiv.org/pdf/1506.03340.pdf [cs.CL].

[3] Danqi Chen et al. *Reading Wikipedia to Answer Open-Domain Questions*. 2017. arXiv: 1704.00051 [cs.CL].

[4] Shuohang Wang and Jing Jiang. *Machine Comprehension Using Match-LSTM and Answer Pointer*. 2016. arXiv: 1608.07905 [cs.CL].

[5] Minjoon Seo et al. *Bidirectional Attention Flow for Machine Comprehension*. 2018. arXiv: https://arxiv.org/abs/1611.01603 [cs.CL].