Question Answering

Ramaseshan Ramachandran

Question answering is a process where AI systems provide answers to questions asked in everyday language

Types of Questions

▶ Extractive QA: Finds the answer directly from the text.

▶ Abstractive QA: Generates a new answer, which might rephrase information

▶ Model Examples: BERT for extractive QA and T5 for abstractive QA

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 Phrase:** The component of a question that specifies the subject of search.

▶ **Question Type:** A categorization of a question based on its intended purpose.

▶ **Answer Type:** A class of objects that the question seeks to identify.

▶ **Question Focus:** The property or entity that the question is inquiring about.

▶ **Question Topic:** The subject or event that the question pertains to.

▶ **Candidate Passage:** Any text retrieved by a search engine in response to a question, ranging from sentences to documents.

▶ **Candidate Answer:** The text ranked based on its perceived suitability as an answer to the question.

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

```
ENTY:animal What fowl grabs the spotlight after the Chinese Year of the M
ABBR:exp What is the full form of .com ?
HUM:ind What contemptible scoundrel stole the cork from my lunch ?
HUM:gr What team did baseball 's St. Louis Browns become ?
HUM:title What is the oldest profession ?
DESC:def What are liver enzymes ?
HUM:ind Name the scar-faced bounty hunter of The Old West .
NUM:date When was Ozzy Osbourne born ?
DESC:reason Why do heavier objects travel downhill faster
...
```

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

```python
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.?>}')
```

Simple rules could be defined to classify questions
For example,

1. $if\ QuestionStartsWith(who)\ or\ QuestionStartsWith(whom)$
   $TopHierarchy \leftarrow$ *HUMAN*
   $Class \leftarrow$ *PERSON*
   fi

2. $if\ QuestionStartsWith(where)$
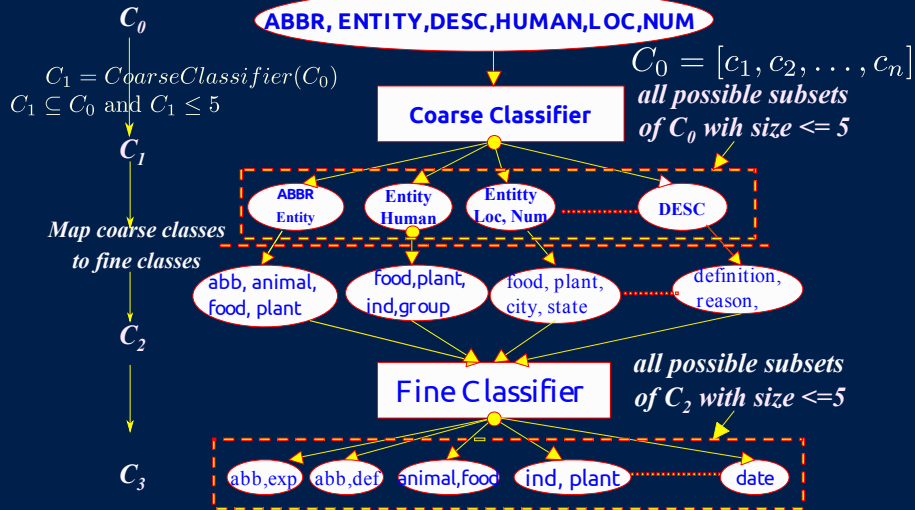 $TopHierarchy \leftarrow$ *LOCATION*
   $Class \leftarrow$ *CITY*
   fi

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

- 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

C_0

$C_1 = CoarseClassifier(C_0)$
$C_1 \subseteq C_0$ and $C_1 \leq 5$

C_1

Map coarse classes
to fine classes

C_2

C_3

ABBR, ENTITY, DESC, HUMAN, LOC, NUM

Coarse Classifier

$C_0 = [c_1, c_2, \ldots, c_n]$
all possible subsets
of C_0 wih size <= 5

ABBR Entity

Entity Human

Entitty Loc, Num

DESC

abb, animal, food, plant

food, plant, ind, group

food, plant, city, state

definition, reason,

Fine Classifier

all possible subsets
of C_2 with size <=5

abb,exp

abb,def

animal,food

ind, plant

date

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 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.
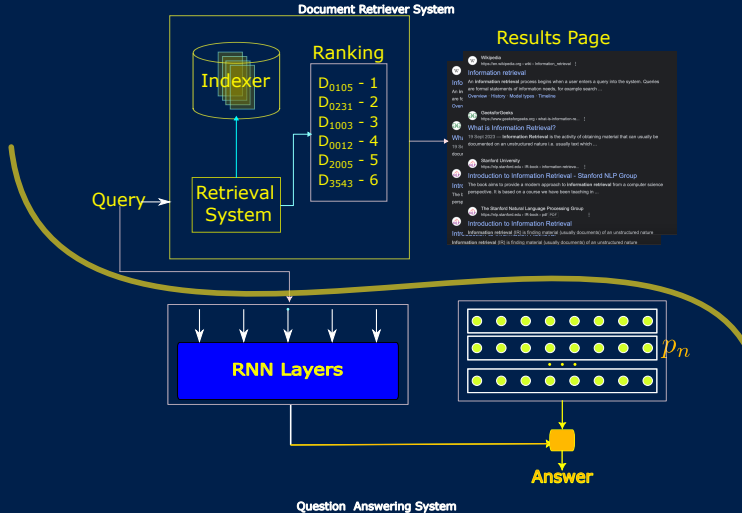
Figure: QA System

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]

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 groundbreaking 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
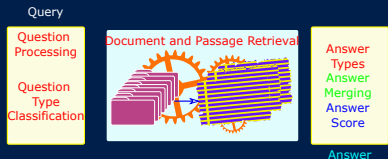
► **Document Retriever**: Utilizes bigram hashing and TF-IDF matching to efficiently return relevant articles based on a given question.

► **Document Reader**: A multi-layer recurrent neural network model trained to extract answer spans from the retrieved documents.

▶ 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

▶ 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

- 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 q_i is output from the hidden unit.
- For all the time states, a weighted sum q and a single embedding of the question is the output - $q = [q_1, q_2, q_3, \ldots q_l]$

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

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

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 $p = (p_1, p_2, \ldots, p_m)$ 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 $f_2 = \mathbf{1}(p_i \in q_i)$
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)

P_{start} P_{end}

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

RNN
(LSTM/GRU)

$\mathbf{p}$

RNN
(LSTM/GRU)

$\hat{p}_i$

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

attention

q_1

POS-NN

PERSON

Word embedding

Who is CV Raman?

CV Raman was an Indian Physicist
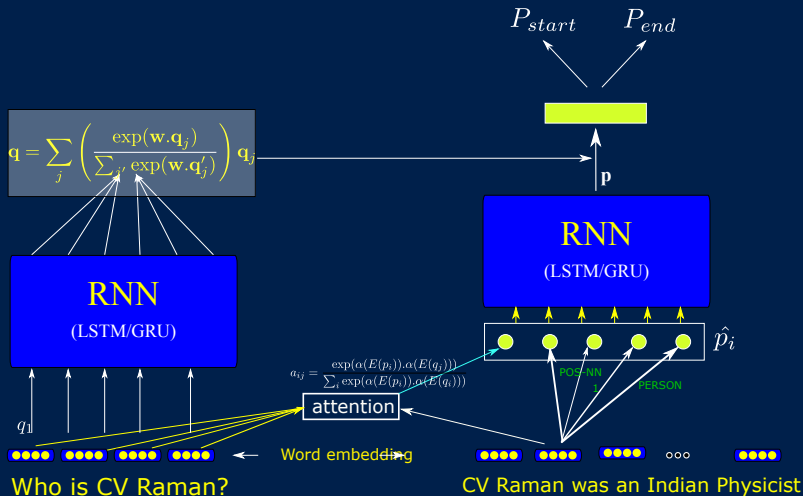
▶ The goal is to predict the span of tokens that is most likely the correct answer

▶ 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:

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

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

▶ 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 = $\arg\max_j \left(P_{start}(i) \times P_{end}(i'), j = 1 \cdots, n \right)$

► 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)

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*

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

	Relevant	Not relevant
Retrieved	TP	FP
Not Retrieved	FN	TN

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

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

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

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

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

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 $H \in \mathbb{R}^{\not{E} \times \mathbb{T}}$ is obtained from context word vectors, X

► Contextual Query vector $U \in \mathbb{R}^{\not{E} \times \mathbb{J}}$ 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.

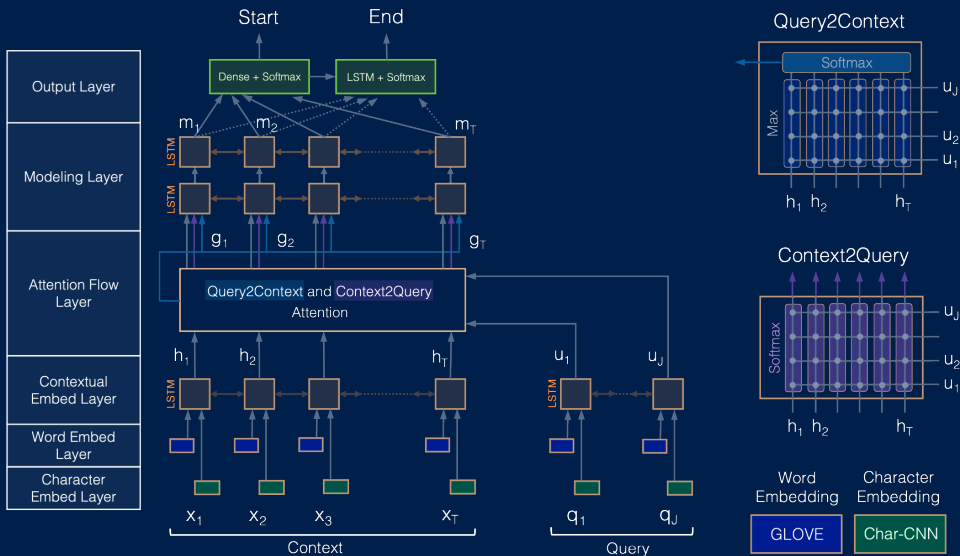► Output Layer: Predicts answers by finding sub-phrases in the paragraph based on start and end indices

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 ⌣
- ▶ `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

▶ Essential for comparing QA system implementations

▶ Primary metrics (Yao, 2014):

    ▶ Accuracy

    ▶ F1 Score

	Actual Positive	Actual Negative
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

▶ Challenge in QA systems:

▶ High TN rates can inflate accuracy scores

▶ Particularly problematic for fact-based questions

Precision

$$P = \frac{TP}{TP + FP}$$

▶ Focus: Correctness of selected answers
▶ Preferred for definition/list QA

Recall

$$R = \frac{TP}{TP + FN}$$

▶ Focus: Completeness of correct answers
▶ Preferred for fact-based QA

$$F1 = \frac{2 \times P \times R}{P + R}$$

▶ Harmonic mean of precision and recall[6], [7], [8]

▶ Addresses precision-recall trade-off

▶ Preferred for systems requiring balance

▶ **Recall-focused:**
  ▶ Fact-based QA systems
  ▶ High TP critical

▶ **Precision-focused:**
  ▶ Definition/list QA systems
  ▶ Minimizing FP essential

▶ **F1 Score:**
  ▶ When balance between P & R needed
  ▶ General system comparisons

[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". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/afdec7005cc9f14302cd0474fd0f3c96-Paper.pdf.

[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: 1611.01603 [cs.CL]. URL: https://arxiv.org/abs/1611.01603.

[6]   Amreen Kaur et al. "A Review on Hindi Question Answering System". In: *International Journal for Modern Trends in Science and Technology* 5 (Nov. 2019), pp. 181–185. DOI: 10.46501/IJMTST051189.

[7]   Xuchen Yao and Benjamin Van Durme. "Information extraction over structured data: Question answering with freebase". In: *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: long papers)*. 2014, pp. 956–966.

[8]   Guangyou Zhou et al. "Improving question retrieval in community question answering using world knowledge.". In: *Ijcai*. Vol. 13. Citeseer. 2013, pp. 2239–2245.