Efficient Natural Language Processing

Question Answering

Fabian Klein

8. February 2012

Table of contents

Definition

- IR / NLP
- automatically answer questions posed in NL
- here: from a text corpus

Factoid Questions

Questions asking for a single answer phrase.

- e.g. "When did the Challenger explode?"
  (28. January 1986)
- "Who was chosen to be the first black chairman of the military Joint Chiefs of Staff?"
  (Collin Powell)

Definition Questions

Question asking for facts about a entity.

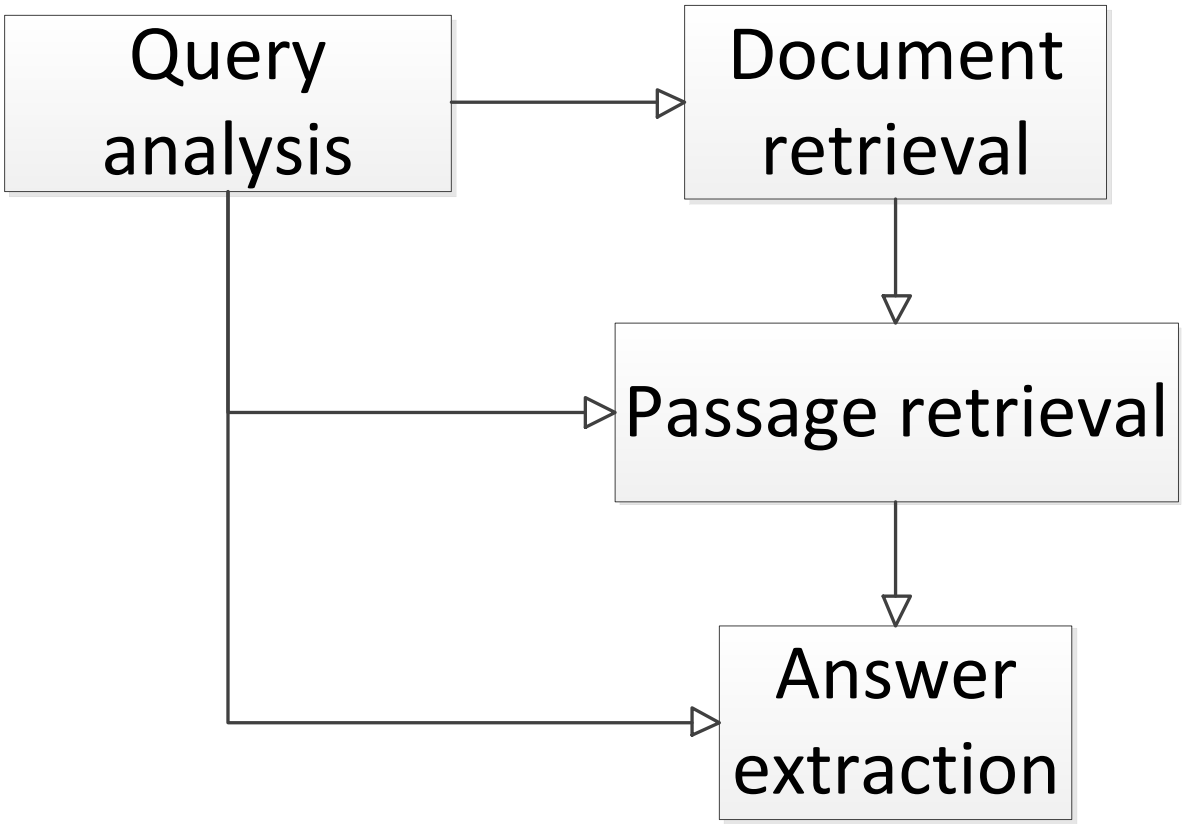
- e.g. "What is Iqra?"
  (Arabic word for read/Gabriel's first word to Mohammed)

List Questions

Question asking for multiple instances.

- e.g. "What Chinese provinces have a McDonalds restaurant?" (Guangdong, Beijing, Shanghai, Jiangsu, ...)

Parts of QA-Systems



Expected Answer Type Using Patterns

answer type	"pattern"	example
PLACE$	Where	In the Rocky Mountains
COUNTRY$	Where/What country	United Kingdom
STATE$	Where/What state	Massachusetts
PERSON$	Who	Albert Einstein
ROLE$	Who	Doctor
NAME$	Who/What/Which	The Shakespeare Festival
ORG$	Who/What	The US Post Office
DURATION$	How long	For 5 centuries
AGE$	How old	30 years old
YEAR$	When/What year	1999
TIME$	When	In the afternoon
DATE$	When/What date	July 4th, 1776
VOLUME$	How big	3 gallons
AREA$	How big	4 square inches
LENGTH$	How big/long/high	3 miles
WEIGHT$	How big/heavy	25 tons
NUMBER$	How many	1,234.5
METHOD$	How	By rubbing
RATE$	How much	50 percent
MONEY$	How much	4 million dollars

Additional Techniques

- Keyword extraction($\rightarrow$ document retrieval)
- Use Depedencies
  "Which actor ...?"
  expected answer type: PERSON$
  using dependencies: person (HYPERNYM of actor (WordNet))
- Dependency parsing/semantic roles ($\rightarrow$ answer extraction)

Keyword expansion

Boolean Querys

search for synonyms too

e.g.
pope $\Rightarrow$ pope OR "Holy Father" OR "Vicar of Christ" OR ...

definition

- set of documents D
- document = set of terms
- inverse document frequency: $idf(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$

IBM Passage Retrieval

score is linear combination of

- "matching keywords"
  sum of idf values of keywords that occur in query and in passage
- "thesaurus match"
  sum of idf values of query words with synomym in passage
- "mismatch words measure"
  sum of idf values of keywords in query and not in passage
- "dispersion measure"
  number of words between matching query terms
- "cluster of words"
  number of words that occur adjacently in question and passage

Pattern based(Definition Questions)

Pattern	Question	Passage
person-hyponym QP	Who is Alberto Tomba?	The doctors also consult with **former Italian Olympic skier** Alberto Tomba, along with other Italian athletes
QP, the AP	What is Bausch & Lomb?	Bausch Lomb, the **company that sells contact lenses**, among hundreds of other optical products, has come up with a new twist on the computer screen magnifier
QP, a AP	What is ETA in Spain?	ETA, a **Basque language acronym for Basque Homeland and Freedom** has killed nearly 800 people since taking up arms in 1968
QP, an AP	Who is Abu Sayyaf?	The kidnappers claimed they are menbers of the Abu Sayyaf, an **extremist Muslim group**, but a leader of the group denied that
AP such as QP	What is TB?	For the hundreds of Albanian refugees undergoing medical tests and treatments at Fort Dix, the news is mostly good: Most are in reasonably good health, with little evidence of **infectious diseases** such as TB

Techniques

- Entity Recognition (match with expected answer type)
- Dependency Parsing
  compare Question and Answer

Answer justification(Idea)

use Predicate logic

QLF: question logical form
ALF: answer logical form
WA: world axioms

$WA \cup ALF \models QLF$

representation in PL

- $noun\_nn(x_i)$
- noun phrase consisting of $noun_1\_nn(x_{k_1}), \ldots, noun_n\_nn(x_{k_n})$
  $nn\_nnc(x_j, x_{k_1}, ..., x_{k_n}) \wedge noun_1\_nn(x_{k_1}) \wedge \cdots \wedge noun_n\_nn(x_{k_n})$
- $verb\_vb(e_j, x_l, x_m[, x_n])$
  - e_j: eventuality
  - x_l: syntactic subject
  - x_m: syntactic direct object
  - x_n: indirect object
- $verb\_vb(e_j)$ $adverb(e_j)$

Example(Axioms)

example: How did Adolf Hitler die? [suicide]

- Question Axiom:

  $\exists e_1, x_1, \ldots, x_4 (manner\_at(e_1) \wedge adolf\_nn(x_2) \wedge hitler(x_3) \wedge nn\_nnc(x_4, x_2, x_3) \wedge die\_vb(e_1, x_4, x_1))$

Example(Axioms)

- Answer Logical Form(by Semantic Role Labeling):

  It was Zhukov's soldiers who planted a Soviet flag atop the Reichstag on May 1, 1945, a day after Adolf Hitler commited suicide.

  $\exists e_1, \ldots e_4, x_1, \ldots, x_{19}(\alpha \wedge day(x_9) \wedge adolf\_nn(x_{10}) \wedge hitler\_nn(x_{11}) \wedge nn\_nnc(x_{12}, x_{10}, x_{11}) \wedge commit\_vb(e_3, x_{12}, x_{13}) \wedge suicide\_nn(x_{13}) \wedge suicide\_vb(x_{13}, x_{19}, x_{12}))$

- $\alpha = it\_prp(x_{14}) \wedge be\_vb(e_1, x_{14}, x_2) \wedge zhukov\_nn(x_1) \wedge \_s\_pos(x_2, x_1) \wedge soldiers\_nn(x_4) \wedge planted\_vb(e_2, x_2, x_3) \wedge sovjet\_jj(x_3) \wedge flag\_nn(x_3) \wedge atop\_in(e_2, x_4) \wedge reichstag\_nn(x_4) \wedge on\_in(e_2, x_8) \wedge may\_nn(x_5) \wedge 1\_nn(x_6) \wedge 1945\_nn(x_7) \wedge nn\_nnc(x_8, x_5, x_6, x_7)$

Example(Axioms)

- Suicide is a manner of killing.

  $\forall e_1(suicide\_nn(e_1) \rightarrow kill\_nn(e_1) \wedge manner\_at(e_1))$
- Suicide is the act of killing yourself.

  $\forall e_1, x_1, x_2(suicide\_vb(e_1, x_1, x_2) \rightarrow$
  $kill\_vb(e_1, x_1, x_2) \wedge yourself\_nn(x_2))$
- To kill is to cause to die.

  $\forall e_1, \ldots, e_3, x_1, \ldots, x_3(kill\_vb(e_1, x_1, x_2) \rightarrow$
  $cause\_vb(e_2, x_1, e_3) \wedge die\_vb(e_3, x_2, x_3))$

Example(Proof)

1	[]	$\neg manner\_at(x_{15}) \vee \neg adolf\_nn(x_2) \vee \neg hitler\_nn(x_3) \vee \neg nn\_nnc(x_4, x_2, x_3) \vee \neg die\_vb(x_{15}, x_4, x_1))$
2	[]	$adolf\_nn(c_{16})$
3	[]	$hitler\_nn(c_{15})$
4	[]	$nn\_nnc(c_{14}, c_{16}, c_{15})$
5	[]	$suicide\_nn(c_{13})$
6	[]	$suicide\_vb(c_{13}, c_9, c_{14})$
7	[]	$\neg suicide\_nn(x_{13}) \vee manner\_at(x_{13})$
8	[]	$\neg kill\_vb(x_{23}, x_1, x_2) \vee die\_vb(x_{23}, x_2, c_{23})$
9	[]	$\neg suicide\_vb(x_{24}, x_1, x_3) \vee kill\_vb(x_{24}, x_1, x_3)$
10	[5, 7]	$manner\_at(c_{13})$
11	[6, 9]	$kill\_vb(c_{13}, c_9, c_{14})$
12	[11, 8]	$die\_vb(c_{13}, c_{14}, c_{23})$
13	[1,10,2,3,4,12]	$\bot$

Search Strategy

- partition into set-of-support and usable Axioms
- SOS contains question + answer axioms
- at least one clause from set-of-support
- prefer hypernym relations

Axioms from WordNet

- $concept(x) \rightarrow synomym(x)$
- $concept(x) \rightarrow hypernym(x)$
- $concept(x) \rightarrow genus(x) \wedge differentia(x)$
- e.g. prophet: "an authoritative person who divines the future"
  $prophet(x) \rightarrow authoritative(x) \wedge person(x) \wedge divine(e, x, y) \wedge future(y)$

Wolfram Alpha

- structured information
- Transform Question to "precise computable internal form"

Wolfram Alpha(Demo)

- "When did the Challenger explode?"
- "challenger explosion"
- "Which actor died on 22 January 2008?"
- "heath ledger"

AskJeeves(Demo)

- "Why do lobsters and crabs turn red when cooked?"
- "How did Adolf Hitler die?"
- "When did the Challenger explode?"
- "When did Lucelly Garcia, former ambassador of Columbia to Honduras, die?"

OpenEphyra

- developed by Nico Schlaefer (Universität Karlsruhe (TH)/Carnegie Mellon University) et al.
- written in Java
- keyword expansion
- google query
- scoring

Performance(without initialisation)

- stromboli
- about 1-10sec per Question
- maybe too much results considered during Filtering (usually 500+)

Component	computation time
answer selection	61.427%
question analysis	23.157%
document retrieval	11.621%
rest	3.794%

Performance(answer selection)

Component	computation time
AnswerPatternFilter	59.931%
AnswerTypeFilter	14.098%
DuplicateFilter	8.935%
FactoidSubsetFilter	5.464%
rest	11.572%

Document Retrieval: Stemming/Lemmatisation

Treat close related words or words with the same morphological root as synonyms

e.g.

root word	words
fish	fishing, fished, fish, fisher
connect	connection, connections, connective, connected, connecting

Search for words with the same roots

Sources

- Logic Form Transformation of WordNet and it's Applicability to Question Answering [Dan I. Moldovan & Vasile Rus]
- LLC Tools for Question Answering [Dan Moldovan, Sandra Harabagiu, Paul Morarescu, Finley Lacatusu, Adrian Novischi, Adriana Badulescu, Orest Bolohan]
- Answer Mining by Combining Extraction Technique with Abductive Reasoning [Sandra Harabagiu, Dan Moldovan, Christine Clark, Mitchell Bowden, John Williams, Jeremy Bensley]
- Quantative Evaluation of Passage Retrieval Algoritms for Question Answering [Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton]