Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Query Construction

Marian Gawron
Question & Answering

Agenda

- Question types
- transformation
    - determine candidates
    - weight & selection weight computation
- techniques for removing Stopstructure
    - Noun phrase
    - Key concept

Question Types

- ˆwhat (is|are|were|does|do|did|should|can)\s
- ˆwho (is|are|was|were|did|do|does)\s
- ˆhow (to|is|do|did|does|can|would|could|should)\s
- ˆwhy (is|do|are|did|were|does)\s
- ˆwhere (is|was|can|are|were|do|does)\s
- ˆwhen (is|was|are|were|do|did|does)\s
- ˆwhich\s

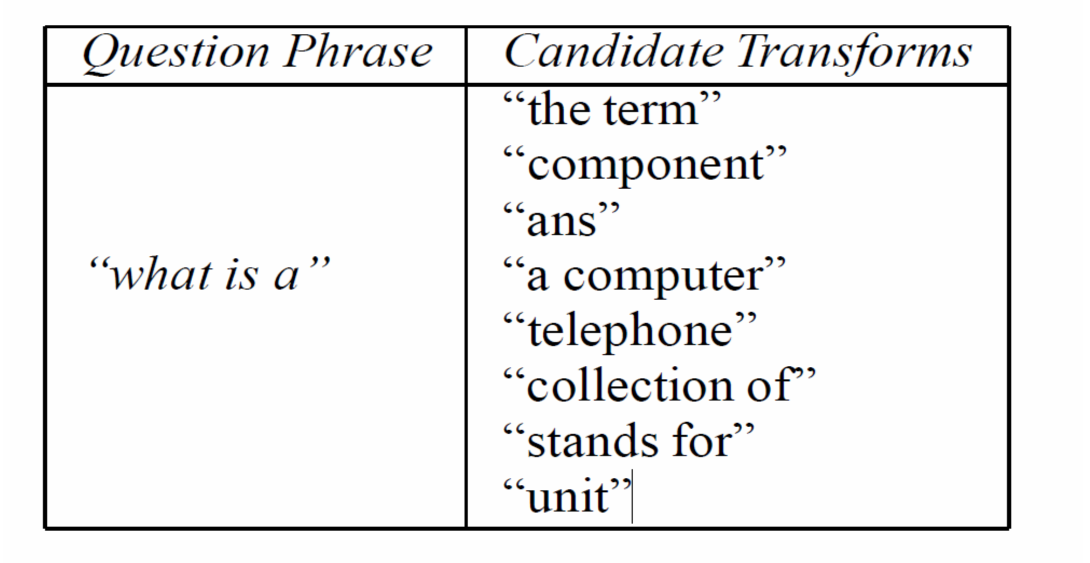
Question Types

- who
- how
- where
- what

- evaluation of phrases
    - „where can I …“   is a where-question
- use of regular expressions to identify categories
    - ^where (is|was|can|are|…) \s

Determine Candidate Transforms - Machine Learning

- <Question, Answer> pair from training set
  - use prefixes of answer



Question Phrase	*Candidate Transforms*
"what is a"	"the term"
	"component"
	"ans"
	"a computer"
	"telephone"
	"collection of"
	"stands for"
	"unit"

- what is a mountain bike? → mountain bike stands for …

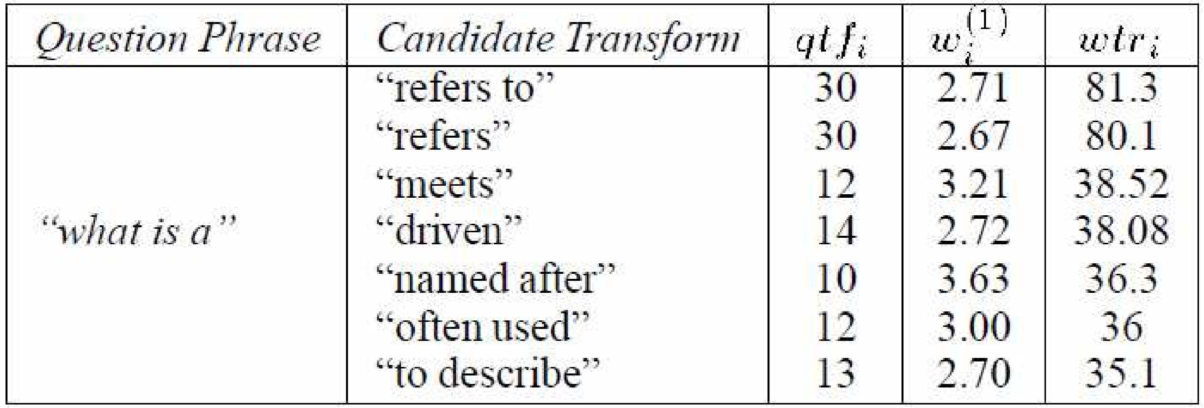
Transformation – weight computation

- candidates for a question category, known from training set
- Robert-Spark Jones term weigth
- w_i = $$\log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)}$$
- for each term t_i
- r= number of relevant docs containing term
- N= total number of docs
- R= number of relevant docs
- n= number of docs containing term

Transformation – selection weight computation

$wtr_i = qtf_i * w_i$

qtf_i = co-occurrence count



Question Phrase	*Candidate Transform*	qtf_i	$w_i^{(1)}$	wtr_i
"what is a"	"refers to"	30	2.71	81.3
	"refers"	30	2.67	80.1
	"meets"	12	3.21	38.52
	"driven"	14	2.72	38.08
	"named after"	10	3.63	36.3
	"often used"	12	3.00	36
	"to describe"	13	2.70	35.1

Transformation – overview

- {Question remaining words} → transformation pipeline
  → {remaining words AND tr_i }

- {“What is a lisp machine?”} → transformation pipeline
  → {(“lisp machine”) AND (“refers to”) }

  - Google results = “is usually”, “usually”
    best ranking in “what is” questions

Transformation – results

- Evaluated 4 question types (what, how, where, who)
- Native Google approach nearly as good as TR-GO results in "what", "how", "where"
- Only "who" questions other approaches work much better

Stopword Removal

- Use lists of stopwords:
  - examine by an inverse document frequency
  - term frequency results similar to idf

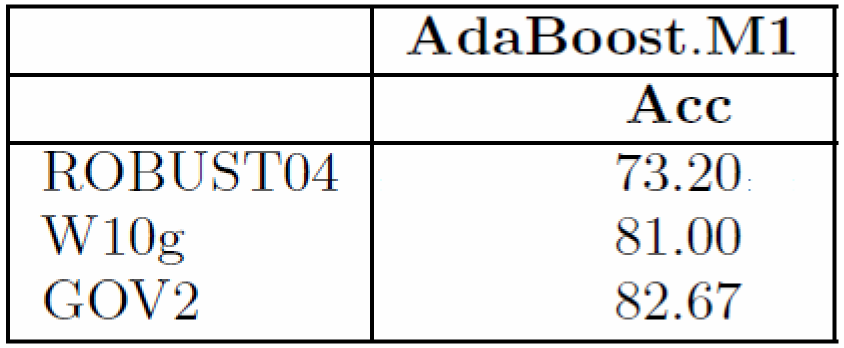
- Problems that occur:
  - “I would like to know the origin of the phrase to be or not to be”
    → “know origin phrase”

Noun phrase detection

- “give me a name for my next horror script”
- Monty Lingua Toolkit to get semantic information
- Noun phrases: “me”, ”a name”, “my next horror script”

Key concept detection

- AdaBoost.M1 meta classifier
- with several features
- GOV2 used to train



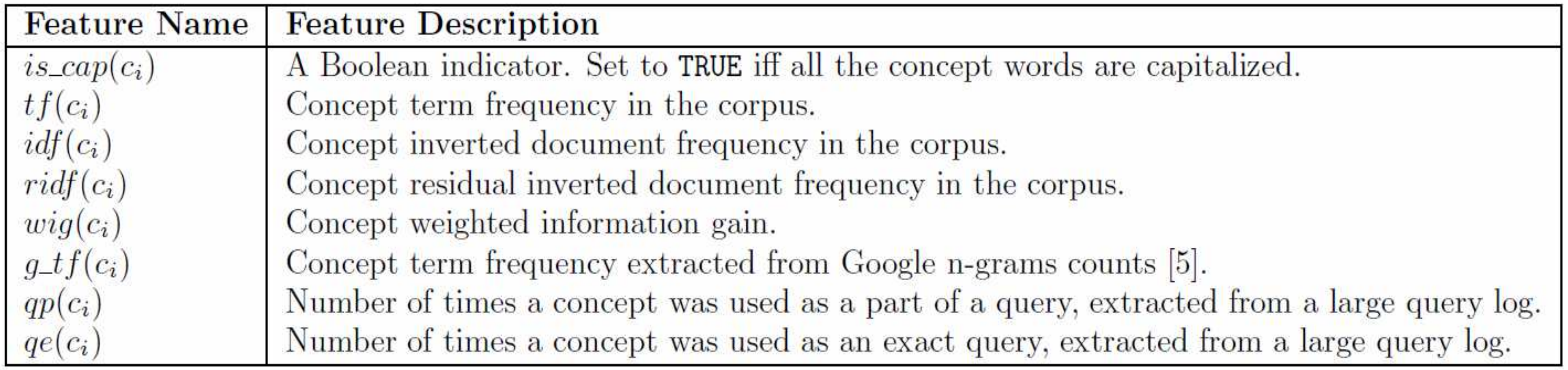
	AdaBoost.M1
	Acc
ROBUST04	73.20
W10g	81.00
GOV2	82.67

- “I read that ions cant have net dipole moments why not”
- “I”, “ions”, “net dipole moments” as Key Concept

Key concepts

Get concepts basically from Noun-Phrases

Concept: single word, multiple words, an idiom



Feature Name	Feature Description
$is\_cap(c_i)$	A Boolean indicator. Set to `TRUE` iff all the concept words are capitalized.
$tf(c_i)$	Concept term frequency in the corpus.
$idf(c_i)$	Concept inverted document frequency in the corpus.
$ridf(c_i)$	Concept residual inverted document frequency in the corpus.
$wig(c_i)$	Concept weighted information gain.
$g\_tf(c_i)$	Concept term frequency extracted from Google n-grams counts [5].
$qp(c_i)$	Number of times a concept was used as a part of a query, extracted from a large query log.
$qe(c_i)$	Number of times a concept was used as an exact query, extracted from a large query log.

Stop Structure

Removal of stop words until first meaningful word

Erase stop sentence which does not provide additional information

“I would like to know the origin of the phrase to be or not to be” → “the phrase to be or not to be”

“… know the origin of the phrase to be or not to be…”

Stop Structure – automatically

- Training collections: GOV, GOV2, WT10G
- Subsets: very short -, short -, long queries

1. Part of speech tag
2. Non-stopword before   and is a stopword
3. Term Frequency, Bi-Term Frequency, Inverse Term Frequency

4. Classifier + features

Results

- Manual Stop Structure much better then original query or Stopword removal
- combined with Stopwords slightly different
- Classified approach nearly as good as manual

Sources

- Eugene Agichtein, Steve Lawrence, and Luis Gravano.
“Learning search engine specific query transformations for question answering”.
In Tenth International World Wide Web Conference (WWW-10), Hong Kong, May 2001.

- S. Huston and W. B. Croft.
“Evaluating verbose query processing techniques”.
In Proc. of SIGIR 2010, pages 291–298.

- M. Bendersky and W. B. Croft.
“Discovering key concepts in verbose queries”.
In SIGIR ’08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pages 491–498, New York, NY, USA, 2008. ACM.