AnswerFinder at TREC 2004

Diego Mollá **Mary Gardiner**
Centre for Language Technology
Macquarie University
Sydney, Australia
{diego,gardiner}@ics.mq.edu.au

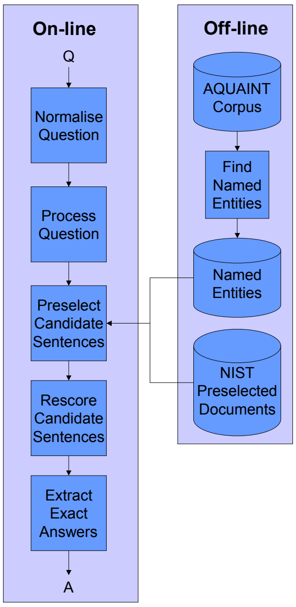
Abstract

AnswerFinder combines **lexical**, **syntactic**, and **semantic** information in various stages of the QA process. The candidate sentences are preselected on the basis of the presence of named entities of the expected answer type, and **a scoring system that combines the overlap of words, grammatical relations, and flat logical forms**. The candidate answers are extracted from the set of compatible named entities and the output of a **logical-form pattern matching algorithm**.

System Overview

The question answering procedure follows a pipeline structure with the following stages:

1. Normalise the question by replacing pronouns with the target text.
2. Process the question and find the expected answer type.
3. Preselect candidate sentences from NIST preselected documents using lexical information (NE information and word overlap).
4. Rescore the candidate sentences using syntactic and semantic information.
5. Extract exact answers using named entities and logical form patterns.



Resources:

- NE data generated by GATE (Gaizauskas et al. 1996)
- NIST preselected documents

Grammatical Relation Overlap (gro)

The grammatical relations (GRs) devised by Carroll et al. (1998) encode the syntactic information of the questions and candidate answer sentences.

GRs are independent of the actual parser used. In our case, we used Connexor FDG (Tapanainen and Järvinen, 1997).

Like dependency structures, GRs are easily incorporated into an overlap-based similarity measure:

Q: *How far is it from Mars to Earth?*

(subj be it _)

(xcomp from be mars)

(ncmod _ be far)

(ncmod _ far how)

(ncmod earth from to)

A: *It is 416 million miles from Mars to Earth.*

(ncmod earth from to)

(subj be it _)

(ncmod from be mars)

(xcomp _ be mile)

(ncmod _ million 416)

(ncmod _ mile million)

The similarity-based score is the number of grammatical relations shared between the question and the sentence. The example above has a similarity score of 2 (shared GRs highlighted)

Flat Logical Form Overlap (lfo)

Semantic information is represented by means of flat logical forms (Mollá 2001). These logical forms use **reification** to flatten out nested expressions in a way similar to other QA systems (e.g. Harabagiu et al. 2001).

The logical forms are produced by means of a process of bottom-up traversal of the dependency structures returned by Connexor FDG.

Flat logical form overlap must account for variations in variable names in the term arguments. We used Prolog unification for this by converting the question variables into Prolog variables.

Example of FLF Overlap

Q: *What is the population of Iceland?*

object(iceland, O6, [X6])

object(population, O4, [X1])

object(what, O1, [X1])

prop(of, P5, [X1, X6])

A: *Iceland has a population of 270000*

dep(270000, d6, [x6])

object(population,o4,[x4])

object(iceland,o1,[x1])

evt(have,e2,[x1,x4])

prop(of,p5,[x4,x6])

Unification:

O4 = o4	X1 = x4
P5 = p5	X6 = x6

FLF Patterns

To account for the differences between the FLF of a question and that of a sentence containing an answer, AnswerFinder uses FLF templates that transform the FLF of a question into that of likely FLFs of answers.

What is X of Y?

object(ObjX,VobjX,[VeX]),

object(what,_,[VeWHAT]),

object(ObjY,VobjY,[VeWHAT]),

prop(of,_,[VexistWHAT,VeX])

Y has a X of ANSWER

dep(ANSWER,ANSW,[VeANSW]),

prop(of,_,[VeY,VeANSW]),

object(ObjX,VobjX,[VeX]),

evt(have,_,[VeX,VeWHAT]),

object(ObjY,VobjY,[VeY]

The process to matching and replacement uses Prolog unification. The resulting FLF has a greater chance of having a higher overlap with the answer sentence FLF. The transformed FLF of *What is the population of Iceland?* has an overlap score with the answer sentence of 5:

dep(ANSWER,ANSW,[VeANSW]),

prop(of,_,[VeY,VeANSW]),

object(iceland,O6,[X6]),

evt(have,_,[X6,X1]),

object(population,O4,[VeY])

Exact Answer Extraction

Exact answer candidates are:

- NEs of the expected answer type.
- Instantiation values of the ANSWER variable of the FLF patterns.

Exact answer candidates are scored on the basis of the score of the candidate sentence they appear. If the same string is an NE and an ANSWER value, then the score is doubled.

Exact answer duplicates are merged by adding up the scores of the individual exact answer candidates.

Final Answer

Factoid
Return the top scoring answer, or "NIL" if all scores are 0.

List
Return all scoring answers over a threshold, or the top scoring candidate sentence if all scores are 0.

"Other"
Same as list after converting the question into *What is <target>?*

Results

Run	Factoid	List	"Other"	Total
answfind1 (lfo,single)	0.10	0.081	0.080	0.090
answfind2 (3gro + lfo, single)	0.10	0.080	0.080	0.090
answfind3 (3gro + lfo, multiple)	0.10	0.080	0.080	0.090

References

J. Carroll, T. Briscoe, A. Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. In *Proc. LREC98.*

R. Gaizauskas, H. Cunningham, Y. Wilks, P. Rodgers, K. Humphreys. 1996. GATE: an environment to support research and development in natural language engineering. In *Proc. 8th IEEE International Conference on Tools with Artificial Intelligence*, Toulouse, France.

S. Harabagiu, D. Moldovan, M. Pasça, M. Surdeanu, R. Mihalcea, R. Gîrju, V. Rus, F. Lăcătuşu, R. Bunescu. 2001. Answering complex, list and context questions with LCC's question-answering server. In E.V. Voorhees, D.K. Harman (eds.), *Proc. TREC 2001.*

D. Mollá. 2001. Ontologically promiscuous flat logical forms for NLP. In H. Bunt, I. van der Sluis, E. Thijsse (eds.), *Proc. IWCS-4*, 249-265. Tilburg University.

P. Tapanainen, T. Järvinen. 1997. A non-projective dependency parser. In *Proc. ANLP 97*. ACL.

Acknowledgements

This work is supported by the Australian Research Council (ARC Discovery project ID DP0450750)