Question Answering for Summarisation

Diego Mollá   Stephen Wan

Centre for Language Technology, Macquarie University, Sydney, Australia

{diego,swan}@ics.mq.edu.au

Abstract

Our approach integrates a question answering system to select the most relevant sentences. We used AnswerFinder, a question answering system developed at Macquarie University. The text of a DUC topic was split into individual sentences, and each sentence was passed to AnswerFinder as a separate question. The sentences returned by AnswerFinder are reranked and collated to produce the final summary. This system will serve as a baseline upon which we intend to develop methods more specific to the task of question-driven summarisation.

From Topics to Sequences of Questions

We used the version of AnswerFinder that participated in the QA track of TREC 2004 [3]. Questions in the QA track of TREC 2004 are grouped into topics such that each topic has questions about specific aspects of the topic. Every DUC topic was converted into a TREC topic by using the same topic name as in the DUC topics, and splitting the DUC topic description (<narr> field) into individual sentences. Each individual sentence was treated as a "list" question by AnswerFinder.

For example, the DUC topic D0602B is:

num	D0602B
title	steroid use among female athletes
narr	Discuss the prevalence of steroid use among female athletes over the years. Include information regarding trends, side effects and consequences of such use.

The resulting TREC topic is:

Number D0602B
Title steroid use among female athletes

D0602B.1	LIST	Discuss the prevalence of steroid use among female athletes over the years.
D0602B.2	LIST	Include information regarding trends, side effects and consequences of such use.

Extracting Candidate Answer Sentences

We modified AnswerFinder so that it returns the sentences that are most likely to contain the answer (rather than finding the exact answer). The resulting system has the following modules:

Question Normalisation

AnswerFinder performs simple anaphora resolution of question strings. In particular, AnswerFinder replaces the pronouns of the question with the topic text. Simple morphological rules were used to ensure that the sentences are grammatical.

Question Classification

AnswerFinder uses a set of 29 regular expressions to determine the expected named entity type. In addition, specific keywords in the questions indicate expected answer types.

Candidate Sentence Extraction

Given the set of documents provided by NIST, AnswerFinder selects 100 sentences from these documents as candidate answer sentences.

Candidate sentences are selected in the following way:

1. The documents provided by NIST are split into sentences.
2. Each sentence is assigned a numeric score: 1 point for each non-stopword overlapping with the question string, and 10 points for the presence of a named entity of the expected answer type.
3. For each question, the 100 top scoring sentences are returned as candidate answer sentences.

Sentence Re-scoring

The 100 candidate sentences are re-scored based on the combination of lexical, syntactic, and semantic features:

Lexical: The combined word overlap and named entity score.

Syntactic: The grammatical relation overlap score.

Semantic: The overlap of flat logical form extended with patterns.

Grammatical Relation Overlap Score

We used [1]'s grammatical relations to encode the syntactic information of questions and candidate answer sentences.

An example of the grammatical relations for question and candidate sentence follows:

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 c relations shared between question and sentence (two in the above example).

Flat Logical Form Patterns

Semantic information is represented by means of flat logical forms [2]. A straightforward way of using the flat logical forms is to compute their overlap in the same way as with grammatical relations:

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])

With the goal to take into consideration the differences between a question and the various forms to answer it, AnswerFinder uses patterns that capture the expected form of the answer sentence and locate the exact answer:

Question Pattern: *What is X of Y?*
object(ObjX,VobjX,[VeX]),
object(what,_,[VeWHAT]),
object(ObjY,VobjY,[VeWHAT]),
prop(of,_,[VexistWHAT,VeX])

Answer Pattern: *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])

As the logical form of *What is the population of Iceland?* matches the above question pattern, then its logical form is transformed into:

Q: *What is the population of Iceland?*
dep(ANSWER,ANSW,[VeANSW]),
prop(of,_,[VeY,VeANSW]),
object(iceland,o6,[x6]),
evt(have,_,[x6,x1]),
object(population,o4,[VeY])

Now the transformed logical form shares all five terms with the logical form of *Iceland has a population of 270000*.

Sentence Combination

Our overall summary building strategy consisted of the following steps:

1. Perform any necessary re-ranking of sentence lists according to the basis of their contribution to the final answer.
2. Pop off the best sentence from each answer set and insert it into the summary portion reserved for the question associated with that list.
3. Repeat step 2 until the summary limit is full.

Re-ranking

A sentence that answered multiple questions had its score boosted. This was achieved by keeping the first instance of the duplicated sentence (in some answer set) but increasing its extraction score (as computed by AnswerFinder) by adding the scores of subsequent repetitions found in later answer sets. These duplicates were then removed.

Selecting the Best Sentences

To flesh out the summary skeleton, we iterated across answer sets in 'question order'. The top sentence was removed from the answer set and then inserted into the appropriate portion reserved for that answer set in the summary skeleton.

We kept track of the best extraction score seen so far, regardless of which question is being answered (the *Recent Best* score). We then iterated across answer sets and if we found an answer that was as good as Recent Best score (i.e.. equal to), we appended it to the end of the relevant portion in the summary. If no sentences are found that were as good as the Recent Best score, we reduced the Recent Best score by one and re-iterated across answer sets. This process of filling in the summary ends when all remaining sentences would exceeds our given word limit.

Evaluation Results

The following table includes the scores of our system, the mean of all the participating systems, the best scores, the worst scores, and the scores of the NIST baseline system.

		Responsiveness		*Automatic Eval.*		
Run	*Quality*	*Content*	*Overall*	*R2*	*SU4*	*BE*
AnswerFinder	3.20	2.40	2.10	0.08	0.13	0.04
Rank out of 34	*21-27*	*24-28*	*20-28*	*9-19*	*17-22*	*9-21*
Mean	3.35	2.56	2.19	0.07	0.13	0.04
Median	3.40	2.60	2.20	0.08	0.13	0.04
Best	4.10	3.10	2.40	0.10	0.16	0.05
Worst	2.30	1.70	1.30	0.03	0.06	0.00
Baseline	4.40	2.00	2.00	0.05	0.10	0.02

References

[1] John Carroll, Ted Briscoe, and Antonio Sanfilippo. Parser evaluation: a survey and a new proposal. In *Proc. LREC98*, 1998.

[2] Diego Mollá. Ontologically promiscuous flat logical forms for NLP. In Harry Bunt, Ielka van der Sluis, and Elias Thijsse, editors, *Proceedings of IWCS-4*, pages 249–265. Tilburg University, 2001.

[3] Diego Mollá and Mary Gardiner. Answerfinder at TREC 2004. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proc. TREC 2004*, number 500-261 in NIST Special Publication. NIST, 2005.

Acknowledgements

This work is supported by the Australian Research Council, ARC Discovery grant DP0450750.