Efficient Question Answering with Question Decomposition and Multiple Answer Streams

Sven Hartrumpf[1], Ingo Glöckner[1], and Johannes Leveling[2]

[1] Intelligent Information and Communication Systems (IICS)
University of Hagen (FernUniversität in Hagen), 58084 Hagen, Germany
[2] Centre for Next Generation Localisation (CNGL)
Dublin City University, Dublin 9, Ireland

Abstract. The German question answering (QA) system IRSAW (formerly: InSicht) participated in QA@CLEF for the fifth time. IRSAW was introduced in 2007 by integrating the deep answer producer InSicht, several shallow answer producers, and a logical validator. InSicht builds on a deep QA approach: it transforms documents to semantic representations using a parser, draws inferences on semantic representations with rules, and matches semantic representations derived from questions and documents. InSicht was improved for QA@CLEF 2008 mainly in the following two areas. The coreference resolver was trained on question series instead of newspaper texts in order to be better applicable for follow-up questions. Questions are decomposed by several methods on the level of semantic representations. On the shallow processing side, the number of answer producers was increased from two to four by adding FACT, a fact index, and SHASE, a shallow semantic network matcher. The answer validator introduced in 2007 was replaced by the faster RAVE validator designed for logic-based answer validation under time constraints. Using RAVE for merging the results of the answer producers, monolingual German runs and bilingual runs with source language English and Spanish were produced by applying the machine translation web service Promt. An error analysis shows the main problems for the precision-oriented deep answer producer InSicht and the potential offered by the recall-oriented shallow answer producers.

1 Introduction

The German question answering (QA) system IRSAW (Intelligent Information Retrieval on the Basis of a Semantically Annotated Web) employs deep and shallow methods. The deep answer producer is InSicht, which transforms documents to semantic representations using a syntactico-semantic parser, draws inferences on semantic representations with rules, matches semantic representations derived from questions and documents, and generates natural language answers from the semantic representations of documents. Specialized modules refine the semantic representations in several directions: resolving coreferences in documents (and questions) and resolving temporal deixis in documents. To provide a robust strategy for difficult text passages or passages mixing text and other elements, four shallow[3] answer producers are employed. (Note that one of them, SHASE, is using the semantic representation in a simple way.) The resulting five streams of answer candidates, which are produced in parallel, are logically validated and merged by RAVE. Based on the results of validation, RAVE scores the answer candidates and selects the final results.

2 Changes of InSicht for QA@CLEF 2008

2.1 Improved Dialog Treatment

In contrast to QA@CLEF 2007, we trained the coreference resolver CORUDIS [1] on a dialog corpus with anaphors in questions, namely the test questions from QA@CLEF 2007. The training set was derived as follows. First, all coreferences (pronoun to NP, less specific NP to more specific NP) were annotated yielding 29 questions from 20 question series with a coreference. Second, as 20 training texts will not deliver good results, additional question series were created by taking every continuous sequence of 1 to 4 questions from the QA@CLEF 2007 questions. (A sequence is discarded for training if an anaphora leads outside the selected sequence.) Information about discourse boundaries (topic starts) was ignored because this kind of information is missing in many applications. Third, the resulting 462 question series were fed into the usual training process of CORUDIS. Note that also the answer to a question could be integrated as a possible antecedent, but as only two QA@CLEF 2007 questions show a coreference to the preceding answer, this was ignored. In 2008, the number of such cases increased to four so that this option has become more relevant.[4]

2.2 Question Decomposition

Question decomposition was systematically added to InSicht for QA@CLEF 2008. A decomposition method tries to simplify complex questions by first asking a *subquestion* whose answer is used to form a *revised question* which is often easier to answer than the original question.[5] For example, question decomposition for *Welches Metall wird zur Goldwäsche benutzt?/Which metal is used for washing gold?* (qa08_192) leads to the subquestion *Nenne Metalle/Name metals* with answers like *Eisen/iron* and *Quecksilber/quicksilver* and revised questions like *Wird Quecksilber zur Goldwäsche benutzt?/Is quicksilver used for washing gold?* Answers to original questions found by decomposition often require support for the answered subquestion and the revised question, i.e. the answer to the original question is supported by sentences from different documents.

[3] i.e. not relying on semantic representations of sentences

[4] For corpus documents, the statistical model trained on newspaper articles is chosen instead of the model from question series.

[5] The term *decomposition* is sometimes used in a different sense when a biographical question like *Who was Bernini?* is broken down into a set of standard questions [2].

To evaluate question decomposition after QA@CLEF 2008, we annotated all German QA@CLEF questions since 2003 with decomposition classes (see [3] for details on the annotation, the decomposition classes, and the decomposition methods). For 2008, 21 questions (10.5%) were annotated as decomposable. This percentage is lower than in previous years: from 2004 till 2007, the percentage was 17.1%. Examples from QA@CLEF 2008 are qa08_044 (*Wieviele Bundesländer hat Österreich?/How many states does Austria have?*) and qa08_192 as discussed above. As expected, some answers (e.g. for qa08_192) were not found when decomposition was turned off.

2.3 Performance Improvement

Adding features to the deep producer InSicht yields better results, but often with a longer runtime. Therefore, several performance improvements were tried. As query expansion by logical rules (applied in backward chaining) expands the search space dramatically, this expansion should be reduced by efficient heuristics that do not eliminate good answers. To this end, statistics on successful rule applications (i.e. combinations of logical rules that led to at least one correct answer) were collected from the test collections of QA@CLEF from 2003 to 2007 and some separate question collections. Restricting query expansion to successful rule combinations turned out to be very effective because results for the QA@CLEF 2008 questions stayed stable while runtime decreased by 56%.

3 Shallow QA Subsystems

In addition to the deep producer, IRSAW now employs four shallow producers of answer candidates: QAP [4], MIRA [5], FACT, and SHASE. The latter two have been added for QA@CLEF 2008. FACT employs a fact database, in which relational triples have been indexed, e.g. name2date_of_death("Galileo Galilei", "8. Januar 1642").[6] Relational triples take the same form as triples used in the MIRA producer. The relational triples have been extracted automatically from various sources, including the PND [6], the acronym database VERA, monetary names from ISO 4217, and appositions from the semantic network representation of the Wikipedia and CLEF-News corpora. To answer a question, the relational triple is determined for a question using a machine learning (ML) approach and keywords from the question are used to fill in one argument position of the triple. Answers are extracted from the other argument position of matching triples. Document sentences containing keywords from the question as well as the exact answer string are returned as support for the answer candidate.

SHASE uses the semantic network representation of both question and document sentences to produce answer candidates. The core node representing an answer node is identified in the question semantic network (i.e. the question

[6] The relation type name2date_of_death is viewed as the first component of the triple. Variants of date formats (for the second argument) are explicitly generated and indexed as well because no normalization takes place at this level, yet.

focus node determined by the syntactico-semantic parser). To find answer candidates, the semantic relations to and from the core node, its semantic sort, and its semantic entity are calculated; see [7] for details on the semantic hierarchies. These features are matched with corresponding features of nodes in the document networks. Matching nodes represent answer candidates: the answer string is generated from the semantic network and the document sentence is returned as answer support.

4 Merging Answer Streams by Validation

The answer candidates in the InSicht stream and the shallow QA streams are validated and merged by RAVE (Real-time Answer Validation Engine), a logic-based answer validator designed for real-time QA. It is crucial for the efficiency of RAVE that no answer must be parsed at query time – computing deep linguistic analyses for hundreds of extracted answer candidates during validation is not realistic in a real-time QA setting. The use of logic in RAVE is therefore restricted to validating support passages, i.e. deciding if a passage contains the requested information. This is the case if the logical representation of the question can be proved from the representation of the passage and from the available background knowledge, a criterion which can be checked independently of the answer candidates. Since the passage representations can be pre-computed, this eliminates the need for parsing during validation. Local validation scores are determined based on shallow and (if available) also logic-based features. Separate models were trained for each producer in order to tailor the validation criterion to the characteristics of each answer stream. Training data was obtained from a system run on the QA@CLEF 2007 questions. The resulting 21,447 answer candidates extracted from 27,919 retrieved passages were annotated for containment of a correct answer. Cross-validation experiments on the training set suggested that bagging of decision trees with reweighting of training examples is suited for the task. The local ML-based scores, which estimate the probability that an answer is correct judging from a specific supporting snippet, are aggregated in order to determine the total evidence for each answer. The aggregation model used by RAVE aims at robustness against duplicated information [8]. By pre-ranking arriving answers based on shallow features and computing improved logic-based scores for the most promising candidates until a given timeout is exceeded, RAVE implements an incremental, anytime validation technique [9]. Answer candidates from InSicht do not require logical validation since they result from a precision-oriented QA technique. Their validation rests on the self-assessment of InSicht and the number of alternative justifications found for the answer. Alternatively, the self-assessment can directly be used as the validation score.

5 Description of Runs

All runs with prefix *fuha081* were generated using the ML-based validation scores for InSicht, whereas the runs with prefix *fuha082* used the self-assessment

Table 1. Results for the German question set from QA@CLEF 2008 (CWS: confidence-weighted score; MRR: mean reciprocal rank; R: right, U: unsupported, X: inexact, W: wrong). For accuracy, only first answers that are right or unsupported are counted as correct. Note that only 199 questions were assessed for *fuha081esde*.

Run	Results						
	#R	#U	#X	#W	Accuracy	CWS	MRR
fuha081dede	45	6	8	141	0.255	0.052	0.297
fuha082dede	46	4	11	139	0.250	0.049	0.296
fuha081ende	28	3	6	163	0.155	0.024	0.240
fuha082ende	28	6	6	160	0.170	0.020	0.226
fuha081esde	19	2	9	169	0.105	0.015	0.157
fuha082esde	17	5	5	173	0.110	0.049	0.296

of InSicht. For bilingual QA experiments, the Promt Online Translator (http://www.promt.com/) was employed to translate the questions from English or Spanish to German. Experience from previous CLEF campaigns suggested that Promt would return translations containing fewer errors than other web services for machine translation (MT), which becomes important when translated questions are parsed. However, we found that Promt employs a new MT service (in beta status) and experiments using translations from other web services had a higher performance [10].

6 Evaluation and Discussion

We submitted two runs for the German monolingual task in QA@CLEF 2008 and four bilingual runs with English and Spanish as source language and German as target language (see Table 1). The syntactico-semantic parser employed in InSicht was used to provide a complexity measure for the German questions by counting the semantic relations in parse results (after coreference resolution). This showed a decrease compared to previous years: 9.05 relations per question on average (2007: 11.41; 2006: 11.34; 2005: 11.33; 2004: 9.84). In the bilingual experiments with English and Spanish, about 60% and 40%, respectively, of the performance (measured in right answers) for monolingual German were achieved. Results may have been better with another MT service.

The evaluation of dialog treatment showed that the coreference resolver performed correctly. The only exceptions are the anaphors in the four questions that referred to the answer of the preceding question. These anaphors were incorrectly resolved because this case was not trained (see Sect. 2.1).

Table 2 shows an error analysis for the deep answer producer InSicht. The analysis is based on problem classes that lead to not finding a correct answer; the same classes were used for our participation in QA@CLEF 2004 [11], except that the new class q.incorrect_coreference (coreference resolution errors for questions) is needed for the question series introduced in QA@CLEF 2007. A random

Table 2. Problem classes and problem class frequencies for QA@CLEF 2008

Name	Description	%
q.error	error related to question side	
q.parse_error	question parse is not complete and correct	
q.no_parse	parse fails	3
q.chunk_parse	only chunk parse result	0
q.incorrect_coreference	a coreference is resolved incorrectly	4
q.incorrect_parse	parser generates full parse, but it contains errors	6
q.ungrammatical	question is ungrammatical	0
d.error	error related to document side	
d.parse_error	document sentence parse is not complete and correct	
d.no_parse	parse fails	12
d.chunk_parse	only chunk parse result	16
d.incorrect_parse	parser generates full parse, but it contains errors	16
d.ungrammatical	document sentence is ungrammatical	2
q-d.error	error in connecting question and document	
q-d.failed_generation	no answer string can be generated for a found answer	1
q-d.matching_error	match between semantic networks is incorrect	1
q-d.missing_cotext	answer is spread across several sentences	7
q-d.missing_inferences	inferential knowledge is missing	32

sample of 100 questions that InSicht answered incorrectly was investigated. For questions involving several problem classes, only the one that occurred in the earlier component of processing was annotated in order to avoid speculation about subsequent errors. Similar to our analysis for QA@CLEF 2004, parser errors on the document side and missing inferences between document and question representations are the two main problems for InSicht.

The performance of the shallow QA subsystem[7] has also been assessed. For the 200 questions, a total number of 36,757 distinct supporting passages was retrieved (183.8 per question). 1,264 of these passages contain a correct answer. For 165 of the questions, there is at least one passage that contains an answer to the question. Since these passages form the basis for answer extraction by the shallow producers, this means that for perfect answer extraction, it would theoretically be possible to answer 165 non-NIL questions correctly (or 175 questions including the NIL case). The extraction performance achieved by the answer producers of the shallow subsystem of IRSAW is shown in Table 3. The following labels are used in the table: *#candidates* (average number of extracted answer candidates per question), *#answers* (average number of right answers per question), *pass-rate* (fraction of the 1,264 correct passages from which a correct answer is extracted), *pass-prec* (precision of answer extraction for correct

[7] This subsystem can be improved as follows. Most shallow producers used the semantic network representation for indexing, i.e. no stemming or stopword removal was applied, but full words were indexed. The tokenization and sentence segmentation underlying the semantic network representations often cause the answer extraction to fail. Finally, the shallow producers have not been trained on the Wikipedia.

Table 3. Extraction performance of shallow answer producers

Producer	Results					
	#Candidates	#Answers	Pass-rate	Pass-prec	#Answered	Answer-rate
FACT	14.38	1.43	0.19	0.57	34	0.21
MIRA	80.09	2.15	0.31	0.32	107	0.65
QAP	1.43	0.02	0.00	0.43	2	0.01
SHASE	80.89	1.15	0.16	0.16	81	0.49
all	176.79	4.74	0.50	0.29	132	0.80

passages), *#answered* (number of questions for which at least one right answer is extracted), and *answer-rate* (answered questions divided by total number of questions with a correct supporting passage, i.e. *#answered*/165 in this case). As witnessed by the *answer-rate* of 0.8 for all shallow producers in combination, the answer candidates extracted by the shallow producers cover most of the correct answers contained in the retrieved passages. While perfect selection from the results of the shallow subsystem would answer 132 non-NIL questions correctly (or 142 including NIL questions),[8] RAVE only made 46 correct selections, which indicates that improvements are necessary:

- RAVE is good at identifying passages that contain an answer, but it often cannot discern right answer candidates found in such passages from wrong extractions. The validator needs better features for relating the answer candidate to the result of validating a supporting passage. Moreover, the rudimentary implementation of some existing features (like the answer type check) must be refined in order to achieve better performance.
- Due to technical problems when the training set was generated, the annotations cover only 151 questions of the 2007 test set and less than 30 definition questions. For better ML results, more questions must be annotated.
- The ML technique proved ineffective, but this problem has been addressed in the meantime: After modifying the induction of decision trees in such a way that the MRR on the training set is maximized, RAVE finds 60 correct answers and 102 correct support passages at top-1 position.

The average time for a complete logical validation, i.e. without a time limit, was 1.48 s per question.[9] Prior to the development of RAVE, logical validation used to be one of the most time-consuming stages of IRSAW, but now it no longer slows down the system response time (19.8 s on average).

7 Conclusion

The QA system IRSAW was successfully improved in several ways for QA@CLEF 2008. Coreference resolution for questions was strengthened by generating suit-

[8] Including InSicht would further increase these numbers because often only a deep producer can deliver correct candidates for questions that require inferences.

[9] Times were measured on a standard PC (2.4 GHz CPU frequency).

able training data. Question decomposition in the deep answer producer InSicht opens interesting ways to a fusion of information from different documents or corpora. Adding two more shallow answer sources proved beneficial for robustness. With increasing system complexity, runtime performance becomes critical, but optimization techniques like parallelization and incremental processing help finding useful answers with response times acceptable for interactive querying. The RAVE prototype shows that applying logic-based validation techniques in a real-time QA setting is possible, but richer features and an improved training set must be provided in the next development phase.

References

1. Hartrumpf, S.: Coreference resolution with syntactico-semantic rules and corpus statistics. In: Proceedings of the Fifth Computational Natural Language Learning Workshop (CoNLL-2001), Toulouse, France (2001) 137–144
2. Harabagiu, S.: Questions and intentions. In Strzalkowski, T., Harabagiu, S., eds.: Advances in Open Domain Question Answering. Volume 32 of Text, Speech and Language Technology. Springer, Dordrecht (2006) 99–147
3. Hartrumpf, S.: Semantic decomposition for question answering. In Ghallab, M., Spyropoulos, C.D., Fakotakis, N., Avouris, N., eds.: Proceedings of the 18th European Conference on Artificial Intelligence (ECAI), Patras, Greece (2008) 313–317
4. Leveling, J.: On the role of information retrieval in the question answering system IRSAW. In: Proceedings of the LWA 2006 (Learning, Knowledge, and Adaptability), Workshop Information Retrieval. Universität Hildesheim, Hildesheim, Germany (2006) 119–125
5. Leveling, J.: A modified information retrieval approach to produce answer candidates for question answering. In Hinneburg, A., ed.: Proceedings of the LWA 2007 (Lernen-Wissen-Adaption), Workshop FGIR. Gesellschaft für Informatik, Halle/Saale, Germany (2007)
6. Hengel, C., Pfeifer, B.: Kooperation der Personennamendatei (PND) mit Wikipedia. Dialog mit Bibliotheken **17**(3) (2005) 18–24
7. Helbig, H.: Knowledge Representation and the Semantics of Natural Language. Springer, Berlin (2006)
8. Glöckner, I.: University of Hagen at QA@CLEF 2008: Answer validation exercise. In: Results of the CLEF 2008 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2008 Workshop, Aarhus, Denmark (2008)
9. Hartrumpf, S., Glöckner, I., Leveling, J.: University of Hagen at QA@CLEF 2008: Efficient question answering with question decomposition and multiple answer streams. In: Results of the CLEF 2008 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2008 Workshop, Aarhus, Denmark (2008)
10. Leveling, J., Hartrumpf, S.: Integrating methods from IR and QA for geographic information retrieval. This volume
11. Hartrumpf, S.: Question answering using sentence parsing and semantic network matching. In Peters, C., Clough, P., Gonzalo, J., Jones, G.J.F., Kluck, M., Magnini, B., eds.: Multilingual Information Access for Text, Speech and Images: 5th Workshop of the Cross-Language Evaluation Forum, CLEF 2004. Volume 3491 of LNCS. Springer, Berlin (2005) 512–521