University of Hagen at QA@CLEF 2008: Efficient Question Answering with Question Decomposition and Multiple Answer Streams

Sven Hartrumpf, Ingo Glöckner, Johannes Leveling
Intelligent Information and Communication Systems (IICS)
University of Hagen (FernUniversität in Hagen)
58084 Hagen, Germany
firstname.lastname@fernuni-hagen.de

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 realizes 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 areas. The coreference resolver was trained on question series instead of newspaper texts in order to be better applicable for follow-up questions in question series. 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 2 to 4, by adding FACT and SHASE.

The answer validator introduced in the previous year was replaced with 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 a machine translation web service. An error analysis showed the main problems for the precision-oriented deep answer producer InSicht and the potential offered by the recall-oriented shallow answer producers.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Linguistic processing*
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process*
H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance evaluation*
I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Semantic networks*
I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Language parsing and understanding*

General Terms

Experimentation, Measurement, Performance

Keywords

Question answering, Deep semantic processing of questions and documents, Follow-up questions, Coreference resolution, Question decomposition, Answer merging

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 relevant 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 answer producers are employed. 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

The deep answer producer InSicht was changed in three main aspects that are described in the following subsections.

2.1 Improved Dialog Treatment

In contrast to last year, we retrained the coreference resolver CORUDIS (Hartrumpf, 2001) 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 subsequence of 1 to 4 questions from the QA@CLEF 2007 questions. Information about discourse boundaries (topic starts) was ignored because this kind of information will not be available in many real-world applications. A subsequence is discarded for training if an anaphora leads outside the selected subsequence. 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, it was not done. After the 2008 campaign, it turned out that the number of such cases increased to 4 in QA@CLEF 2008 (qa08_048, qa08_050, qa08_053, and qa08_106) so that this option seems to become more relevant. When CORUDIS is applied to corpus documents, the statistical model trained on newspaper articles is chosen instead of the model from question series.

2.2 Question Decomposition

Question decomposition was systematically added to InSicht for QA@CLEF 2008. A decomposition method tries to simplify answering complex questions by first asking a *subquestion* whose answer is used to form a *revised question* which is often more easy to answer than the original question.[1] 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 the revised question *Wird Quecksilber zur Goldwäsche benutzt?/'Is quicksilver used for washing gold?'* Note that answers to original questions found by question decomposition often require support for the answered subquestions and the revised question, i.e. the answer to the original question is supported by sentences from different documents.

To evaluate question decomposition after QA@CLEF 2008, we annotated all German QA@CLEF questions since 2003 with decomposition classes (see Hartrumpf (2008) for details on the annotation, the

[1] Note that 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, see for example Harabagiu (2006).

decomposition classes, and the decomposition methods). For 2008, 21 questions (10.5%) were annotated as decomposable. This percentage is lower than in previous years; for example from 2004 till 2007, the percentage was 17.1%. Examples from QA@CLEF 2008 are qa08_044 (*Wieviele Bundesländer hat Österreich?*)[2] and question qa08_192 as discussed above. But as expected, some answers (e.g. for question 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, the search space should be reduced by some 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. When restricting query expansion to successful rule combinations, results for the QA@CLEF 2008 questions stayed stable while runtime decreased by 56%. This simple technique turned out to be very effective.

3 Shallow QA Subsystems

In addition to the deep producer, IRSAW now employs four shallow producers of answer candidates: QAP (Leveling, 2006), MIRA (Leveling, 2007), FACT, and SHASE. The latter two have been added for QA@CLEF 2008. FACT makes use of a fact database in which relational triples have been indexed, e.g. name2date_of_death("Galileo Galilei", "8. Januar 1642").[3] 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 (Hengel and Pfeifer, 2005), the acronym database VERA, monetary names from ISO-34217, 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 focus node determined by the syntactico-semantic parser). To find answer candidates, the semantic relations for the core node, its semantic sort, and its semantic entity are calculated; see Helbig (2006) for more details on the semantic hierarchies. These features are matched with the corresponding features of nodes in the document semantic networks. Matching nodes represent answer candidates: the answer string is extracted from the semantic network representation and the document sentence is returned as answer support.

4 Merging Answer Streams by Validation

4.1 Overview of the RAVE Validator

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 ever be parsed at query time – computing a deep linguistic analysis for hundreds of extracted answer candidates during validation is not realistic in the real-time QA setting. This problem is solved by using logic only for validating support passages, i.e. for

[2] The correct answer could also be found directly without decomposition.

[3] The relation type name2date_of_death is viewed as the first component of the triple. Other common date formats are explicitly generated and indexed as well because no normalization takes place at this level, yet.

deciding if the given passage contains a correct answer at all. Establishing the relationship between the considered answer candidate and the information obtained from the question-passage proof is not part of the logic-based processing. Therefore a logical analysis of the answer candidate is no longer needed for validation. Moreover, one question-passage proof is sufficient per passage. The deep features determined by proving the question from the support passage can be re-used if several answer candidates were extracted from the same passage. The parses of the sentences needed for the question-passage proofs are computed before indexing and fetched along with retrieving the support passages.

Local validation scores are determined by an ML method applied to 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. Notice that both a combined model (using the full set of deep and shallow features) and a shallow-only model was generated for each stream. Training data was obtained from a run of the system on the QA@CLEF 2007 test set for German. A total of 21,447 answer candidates extracted from 27,919 retrieved passages were annotated as the basis for machine learning. Preparatory experiments based on cross-validation on the training set have shown that bagging of decision trees with reweighting of training examples is particularly 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 then aggregated in order to determine the total evidence for each answer candidate. RAVE uses a novel aggregation model for that purpose, which aims at robustness against duplicated information; see Glöckner (2008) for a detailed description of the validation approach and the aggregation model.[4]

4.2 Real-Time Validation Approach

RAVE uses an anytime validation technique based on incremental processing of the available answer streams. In order to implement incremental validation, RAVE maintains a priority queue for retrieved passages and a second priority queue for pending answer candidates. The processing loop of RAVE works as follows: If more than $t = 100$ ms have elapsed since the last attempt of reading new passages or new answer candidates, then RAVE reads all passages and answer candidates that have arrived in the meantime. By applying a model learned from passage annotations to the shallow passage features, each passage is assigned a quality score which estimates the probability that the passage contains a correct answer, and the shallow features are cached. Passages with a deep parse are added to the priority queue for later logic-based processing.

The newly arriving answer candidates are first checked for violation of sanity checks. If an answer passes the test, a score based on shallow features is computed. If the supporting passage from which the answer candidate was extracted has a deep parse, then the candidate is added to the priority queue of answer candidates awaiting deep validation. If the supporting passage has no deep parse, however, then the score based on shallow features is directly aggregated, provided that it exceeds a given quality threshold. Having integrated all newly arriving data, the system fetches the answer candidate with the best shallow score from the queue of answer candidates. If no logic-based features are already cached for the corresponding support passage yet, the passage is subjected to logical processing by trying to prove the question literals from the logical representation of the passage and storing the resulting logic-based features. Then a 'deep' answer score based on the combination of deep and shallow features is computed and the result is aggregated if it exceeds the quality threshold. In the event that the priority queue of answer candidates is empty, the available time is utilized by pre-computing logic-based features for the best item in the priority queue of support passages. This will speed up the later evaluation of answer candidates extracted from this passage since the logic-based features can then be fetched from the cache.

This incremental reading/processing loop is repeated until all streams are exhausted or until the specified time limit is exceeded. The system then iterates over all remaining items in the priority queue of answer candidates. If logic-based features are already cached for such a remaining answer candidate, the candidate is assigned a deep score; otherwise only shallow features are used. In any case, the validation

[4] The version of RAVE used for the submitted runs was still lacking the test for compatibility of measurement units of question and answer and the test for fulfillment of temporal restrictions described in the reference. Two additional features were incorporated for the QA@CLEF runs that were not available in the Answer Validation Exercise, viz *irScore* (retrieval score for the supporting passage determined by the IRSAW retrieval system) and *producerScore* (a quality score assigned by each answer producer when generating an answer candidate).

Table 1: Results for the German question set from QA@CLEF 2008 (CWS: confidence-weighted score; MRR: mean reciprocal rank). Note that only 199 questions were assessed for *fuha081esde*.

Run	Results					
	#Right	#Unsupported	#Inexact	#Wrong	CWS	MRR
fuha081dede	45	6	8	141	0.05210	0.29706
fuha082dede	46	4	11	139	0.04868	0.29608
fuha081ende	28	3	6	163	0.02369	0.24041
fuha082ende	28	6	6	160	0.01987	0.22619
fuha081esde	19	2	9	169	0.01541	0.15672
fuha082esde	17	5	5	173	0.04868	0.29608

score is aggregated provided that it exceeds the quality threshold. Finally, RAVE determines the three distinct answers with the highest aggregated score. These answers are returned together with the best supporting snippet found for each answer. If there is no aggregated evidence for any answer at all, then a NIL answer with zero confidence is generated.

4.3 Validation of InSicht Results

Answer candidates generated by InSicht are always directly aggregated – these answers result from a precision-oriented QA technique and do not require logical validation. Since InSicht works independently of the passage retrieval step, RAVE lacks the morpho-lexical information needed for computing shallow passage features and assigning a validation score in the usual way. Two methods for assigning validation scores to the InSicht results were tried. The first was training an ML-based classifier using a special set of features for InSicht, which consists only of the *producerScore* assigned by InSicht itself and an *occurrences* count for the number of alternative justifications that InSicht has found for the answer. The second method was directly using the self-assessment of InSicht, i.e. the *producerScore*, as the local validation score for answer candidates contributed by InSicht.

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 of InSicht. For bilingual QA experiments, the Promt Online Translator[5] was employed to translate the questions from English or Spanish to German. From experience in previous CLEF campaigns, it was expected that this web service would return translations containing fewer errors than other comparable web services for machine translation, which becomes important when deep NLP is applied, i.e. when the translated questions are parsed. However, we found that, in this year, Promt offers a new machine translation service (in beta status) and experiments using translations from other web services had a higher performance, see Leveling and Hartrumpf (2008).

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 an approximate 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).

[5] http://www.promt.com/

Table 2: Problem classes and problem class frequencies for QA@CLEF 2008 (percentages sum to 100.3 due to rounding)

Name	Description	%
q.error	error on question side	
q.parse_error	question parse is not complete and correct	
q.no_parse	parse fails	4.1
q.chunk_parse	only chunk parse result	0.0
q.incorrect_coreference	a coreference is resolved incorrectly	5.4
q.incorrect_parse	parser generates full parse result, but it contains errors	6.8
q.ungrammatical	question is ungrammatical	0.0
d.error	error on document side	
d.parse_error	document sentence parse is not complete and correct	
d.no_parse	parse fails	12.2
d.chunk_parse	only chunk parse result	14.9
d.incorrect_parse	parser generates full parse result, but it contains errors	13.5
d.ungrammatical	document sentence is ungrammatical	1.4
q-d.error	error in connecting question and document	
q-d.failed_generation	no answer string can be generated for a found answer	1.4
q-d.matching_error	match between semantic networks is incorrect	1.4
q-d.missing_cotext	answer is spread across several sentences	5.4
q-d.missing_inferences	inferential knowledge is missing	33.8

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 machine translation service for QA@CLEF 2008.

The evaluation of dialog treatment for the 2008 questions showed that the coreference resolver performed correctly, with one exception: The anaphors in the four questions that referred to the answer of the preceding question were incorrectly resolved because this case was not allowed in the trained coreference model (see Sect. 2.1).

Table 2 contains an error analysis for the deep answer producer InSicht with a predefined classification. The classes are problem classes that lead to not finding the correct answer; the same classes were used for our first participation, QA@CLEF 2004 (Hartrumpf, 2005), 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 sample of 74 questions that InSicht answered incorrectly were investigated. If several problem classes were visible for a question, 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, missing inferences (between document and question representations) and parser errors on the document side are the two main problems for InSicht.

The performance of the shallow QA subsystem 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, i.e. the precision of passage retrieval is 3.44%. 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 MIRA, QAP, FACT and SHASE, this means that for perfect answer extraction and validation, it would theoretically be possible for the shallow subsystem to answer 165 non-NIL questions correctly (or 175 questions including the NIL case). More details on the number of available correct passages for each question are shown in Figure 1.

The actual extraction performance achieved by the answer producers of the shallow subsystem of IR-SAW has also been investigated, see 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), *precision* (correctness rate of answer extraction, i.e. *#answers*/*#candidates*), *pass-rate* (fraction of the 1,264 correct passages from which a correct answer is extracted), *pass-prec* (precision of answer

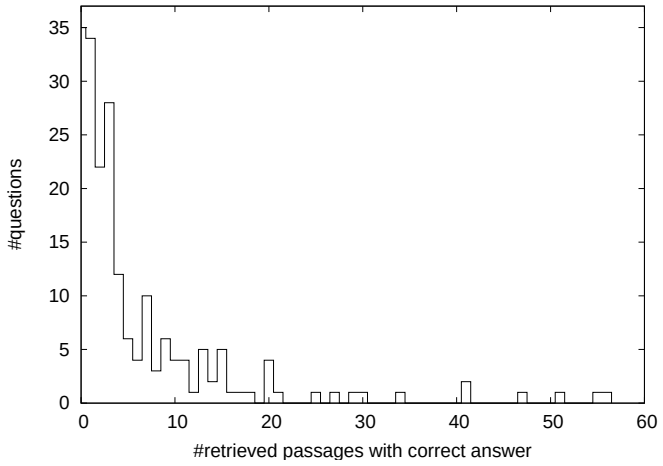

Figure 1: Number of questions with a given number of retrieved passages that answer the question

extraction for correct 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. However, the precision of answer extraction is very low (only 3% when considering all producers), and even for those passages that contain an answer, the precision of extraction (i.e., *pass-prec*), is only 29%. While the strong recall-orientedness of the shallow subsystem provides a good basis for answer selection, the very low precision also means that the burden of spotting the correct answers is shifted to the validation component.

Assuming perfect validation, it would be possible to answer 132 non-NIL questions correctly based on the results of the shallow subsystem (or 142 if one includes the NIL questions). There are even more correct answers if InSicht is also taken into account. However, subsequent processing by RAVE only resulted in 46 correct answers in the best submitted run (*fuha082dede*). This clearly demonstrates that improvements of the validator are necessary. While some minor bugs of the validator have already been fixed (see description of recent changes in Glöckner (2008)), three other problems must be addressed in the near future:

- The first problem is the lack of features which relate the answer candidate to the result of the question-passage proof. As a consequence, RAVE is good at identifying passages which contain an answer, but it often cannot discern right answer candidates (extracted from such a passage) from wrong answer candidates.
- Another problem is the incomplete implementation of the answer-type test which checks the compatibility of the expected answer type of the question and the found answer type. This test is limited

Table 3: Extraction performance of shallow answer producers

Producer	Results						
	#Candidates	#Answers	Precision	Pass-rate	Pass-prec	#Answered	Answer-rate
MIRA	80.09	2.15	0.03	0.31	0.32	107	0.65
QAP	1.43	0.02	0.01	0.00	0.43	2	0.01
SHASE	80.89	1.15	0.01	0.16	0.16	81	0.49
FACT	14.38	1.43	0.10	0.19	0.57	34	0.21
all	176.79	4.74	0.03	0.50	0.29	132	0.80

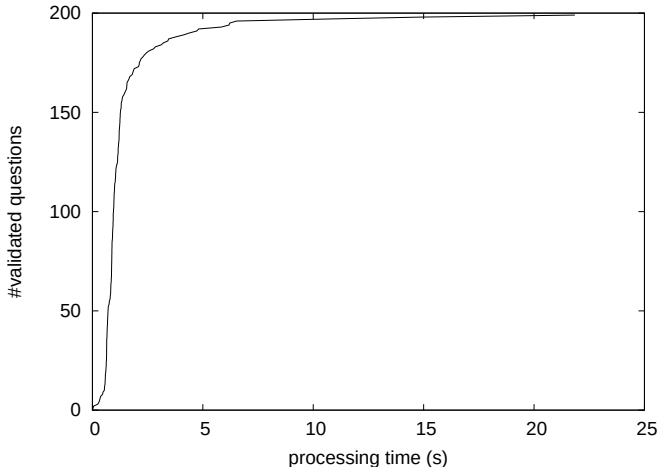

Figure 2: Processing times of RAVE for a complete logical validation

to a few special answer types at the moment, and it is even more restricted for support passages with a failed parse.

- The third problem is concerned with the training set of RAVE. Due to unstable operation of IRSAW when the training set was generated, the annotations cover only 151 questions of the QA@CLEF07 test set and less than 30 definition questions. In order to provide a suitable basis for machine learning, a considerable increase in the number of annotated questions is necessary. Moreover the training data for SHASE is not representative of the current version of the producer since SHASE was apparently broken when generating the training set and hardly produced any correct answers. Therefore the machine learning approach did not result in a useful model for SHASE.

Despite this need for additional validation features and a larger, up-to-date training set, the observed processing times for RAVE confirm that the validator is suitable for application in real-time QA. The processing times for a complete logical validation, i.e. without using a time limit, are shown in Figure 2. The average time needed for answer validation and selection is 1.48 seconds per question.[6] This process involved an average of 79 question-passage proofs per question, which cover all parseable snippets retrieved by IRSAW. Notice that the complete logical validation takes less than 0.95 seconds for half of the questions, and less than 2.45 seconds for 90% of the questions. By specifying a time limit, these processing times can be constrained even further.

7 Conclusion

The QA system IRSAW was successfully improved in several ways for QA@CLEF 2008. Coreference resolution for questions was strengthened by generating suitable training data. Question decomposition in the deep answer producer InSicht opens interesting ways to a fusion of information from different documents or corpora. With increasing system complexity, runtime performance becomes critical, but specialized optimization techniques allow to provide useful answers in near real-time. The latter aspect is still open for improvements in the future, especially with the advent of computers with more and more CPU cores. Adding two more shallow answer producers turned out beneficial for robustness, although the integration in the validator must be improved further. The first prototype of the RAVE answer validator already demonstrates that logic-based processing and real-time answer validation can be reconciled, but additional validation features and an improved training set must be provided in the next development phase.

[6] Processing times were measured by running RAVE in a single thread on an Athlon64X2 4800+ CPU with 2.4 GHz clock rate.

References

Glöckner, Ingo (2008). 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.

Harabagiu, Sanda (2006). Questions and intentions. In *Advances in Open Domain Question Answering* (edited by Strzalkowski, Tomek and Sanda Harabagiu), volume 32 of *Text, Speech and Language Technology*, pp. 99–147. Dordrecht: Springer.

Hartrumpf, Sven (2001). Coreference resolution with syntactico-semantic rules and corpus statistics. In *Proceedings of the Fifth Computational Natural Language Learning Workshop (CoNLL-2001)*, pp. 137–144. Toulouse, France. http://www.aclweb.org/anthology/W01-0717.

Hartrumpf, Sven (2005). Question answering using sentence parsing and semantic network matching. In *Multilingual Information Access for Text, Speech and Images: 5th Workshop of the Cross-Language Evaluation Forum, CLEF 2004* (edited by Peters, Carol; Paul Clough; Julio Gonzalo; Gareth J. F. Jones; Michael Kluck; and Bernardo Magnini), volume 3491 of *Lecture Notes in Computer Science*, pp. 512–521. Berlin: Springer. http://dx.doi.org/10.1007/11519645_50.

Hartrumpf, Sven (2008). Semantic decomposition for question answering. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI)* (edited by Ghallab, Malik; Constantine D. Spyropoulos; Nikos Fakotakis; and Nikos Avouris), pp. 313–317. Patras, Greece.

Helbig, Hermann (2006). *Knowledge Representation and the Semantics of Natural Language*. Berlin: Springer. http://www.springer.com/computer/artificial/book/978-3-540-24461-5.

Hengel, Christel and Barbara Pfeifer (2005). Kooperation der Personennamendatei (PND) mit Wikipedia. *Dialog mit Bibliotheken*, 17(3):18–24.

Leveling, Johannes (2006). 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*, pp. 119–125. Hildesheim, Germany: Universität Hildesheim. http://web1.bib.uni-hildesheim.de/2006/fgir2006/Leveling.pdf.

Leveling, Johannes (2007). A modified information retrieval approach to produce answer candidates for question answering. In *Proceedings of the LWA 2007 (Lernen-Wissen-Adaption), Workshop FGIR* (edited by Hinneburg, Alexander). Halle/Saale, Germany: Gesellschaft für Informatik.

Leveling, Johannes and Sven Hartrumpf (2008). University of Hagen at GeoCLEF 2008: Combining IR and QA for geographic information retrieval. In *Results of the CLEF 2008 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2008 Workshop*. Aarhus, Denmark.