An Integrated Machine Learning and Case-Based Reasoning Approach to Answer Validation

Ingo Glöckner
Faculty for Mathematics and Computer Science
FernUniversität in Hagen
58084 Hagen, Germany
ingo.gloeckner@fernuni-hagen.de

Karl-Heinz Weis
Faculty for Computer Science – Artificial Intelligence Working Group
University of Koblenz-Landau
56016 Koblenz, Germany
khweis@weis-consulting.de

***Abstract*—We propose a case-based reasoning (CBR) approach to answer validation/answer scoring and reranking in question answering (QA) systems, where annotated answer candidates for known questions provide evidence for validating answer candidates for new questions. The use of CBR promises a continuous increase in answer quality, given user feedback that extends the case base. In the paper, we present the complete approach, emphasizing the use of CBR techniques, namely the structural case base, built with annotated MultiNet graphs, and corresponding graph similarity measures. We cover a priori relations to experienced answer candidates for former questions. We describe the adequate structuring of the case base and develop appropriate similarity measures. Finally we integrate CBR into an existing framework for answer validation and reranking that also includes logical answer validation and a shallow linguistic validation, using a learning-to-rank approach for the final answer ranking based on CBR-related features. In our experiments on QA@CLEF questions, the best learned models make heavy use of CBR features. The advantage already achieved by CBR will increase with time due to the automatic improvement with new user annotations given by relevance feedback.**

I. Introduction

Question answering (QA) systems aim at providing concise answers to questions given a document collection.[1] The QA process is often divided into generation of answer candidates, using information retrieval techniques, followed by a validation and selection step that determines the final ranking. The quality of this ranking is crucial since users typically inspect only a few top-ranked results. In this paper we start from a working solution for answer validation and reranking [1] based on deep linguistic analysis and logical reasoning. A learning-to-rank approach determines a numerical estimate of answer quality used for ranking [2]. However, for best results, logical validation needs a comprehensive knowledge base. This is unrealistic in an open-domain setting, and so we would like to include other knowledge sources, such as experience knowledge provided by user annotations. Also the learning approach implemented in the system is not incremental. In order to address these problems, we extend the QA system of interest by a case-based reasoning (CBR) approach, letting answer candidates for known questions provide evidence for validating answer candidates for new questions. This promises a continuous increase in answer quality, if the users of the QA system can provide feedback which extends the case base.

The idea of using CBR technology for question answering is not new. However, its application has typically been limited to FAQ answering from a given list of question/answer pairs [3], [4], [5], using techniques for textual CBR [6].[2] By contrast, we aim at utilizing user annotations for supporting a QA system in better answering unseen questions as well. Instead of directly retrieving answers to the considered question, we thus propose the use of CBR as a new tool for answer validation. Moreover, we aim at a joint framework that integrates CBR with proven techniques for answer validation (such as logical validation) by means of machine learning (ML).

The paper is organized as follows. Sect. II sketches the LogAnswer QA system, our testbed for the CBR extension. It also introduces the graph representations for linguistic meanings on which the CBR will operate. Sect. III introduces the CBR approach for answer validation. Sect. IV explains how the CBR approach can be integrated into an existing learning-to-rank framework by adding CBR-related features. Sect. V uses data from QA@CLEF tasks for evaluating case retrieval and the achieved reranking quality.

II. System Overview

Since our CBR approach extends the LogAnswer QA system [1], [8], we sketch the overall system architecture. LogAnswer is an open-domain QA system for German available on the web. Its knowledge base was derived from textual sources, namely the German Wikipedia and a corpus of newspaper articles. Answers are produced using deep linguistic processing and automated theorem proving [1]. All in all about 12 million sentences have been parsed into semantic networks [1], using the WOCADI parser [9] for linguistic analysis.

Given a question, the system first retrieves pre-analyzed semantic representations of candidate sentences from its retrieval backend. Through answer validation, it then tries to identify those candidate sentences that contain a correct answer to the question. There is a logic-based check, accepting an answer as correct if the logical form of the parsed question can

[1] The annual CLEF evaluation (http://www.clef-initiative.eu/) and the TREC initiative (http://trec.nist.gov/) feature (or have featured) QA tracks, and a wealth of papers on QA systems and evaluation are offered by these websites.

[2] Fouque et al [7] also used CBR for query answering on structured data.

be proved from the logical representation of the considered answer sentence and general background knowledge. Robustness is gained by allowing query relaxation [1] and adding shallow linguistic features. A reranking model based on rank-optimizing decision trees [2] estimates the correctness probability of each answer candidate from the answer validation features. The five top-ranked answers are shown to the user.

LogAnswer is also equipped with a feedback mechanism, allowing users to annotate each displayed result as either correct or incorrect. It is this growing experience source of user annotations that we wish to exploit using CBR.

LogAnswer relies on multilayered extended semantic networks (MultiNet) [10] as a semantic representation for natural language expressions. The concepts representations (nodes in the network with their relationships to other nodes) are enriched by multi-dimensional descriptions in terms of so-called layer attributes [10] (e.g. for capturing quantification). For our purposes, the layer attribute data will also be treated as part of the MultiNet graph. The choice of MultiNet as the semantic representation for LogAnswer was a pragmatic decision since it provides us with a proven toolchain for translating German texts into a semantic representation suitable for knowledge processing. The formalism has been used in applications such as NL interfaces to databases and semantic-based readability checking [10]. It fits well with the idea to develop a CBR system with an optimised graph-based similarity measure to improve our existing answer validation model.

III. CBR Approach for MultiNet Graph Classification

We enhance answer validation by using experience knowledge in the form of cases in a case base. This requires representing the experience knowledge in appropriate data structures. We further define a priori relations of current questions and answer candidates to already experienced answer candidates of former questions. The resulting system module is thus designed as a learning system and based on a Case Based Reasoning control structure ([11], [12], which addresses this problem solving task in particular). CBR can handle sparse, noisy, or even absent information by retrieving the most similar case, if a complete match is not available in the case base.

The case base is defined as in [13] with a case characterization part and a lesson part. We use a structural approach with a common structured vocabulary [11], [13]. The vocabulary container consists of XML describing MultiNet graphs [10].

The explicit knowledge source of the CBR module [12] is expert knowledge in the form of retrieved MultiNet graphs, bulk loaded and parsed from XML input files.

A. Definition of the Case Base

Case bases are defined as in [13], using MultiNet data [10].

Definition 1: (Space of Experience Characterization Descriptions) The *space of experience characterization descriptions* $D = Q \times A$ is the set of possible characterizations for experience items, Q MultiNet graphs of questions (A: of answer candidates).

Definition 2: (Space of Lessons) The *space of possible lessons* recorded in experience items, L, is a Boolean signaling if the answer candidate contains a correct answer passage [2].

Definition 3: (Case, General Definition) A *case* is a pair $c = (d, l) \in D \times L$.

Definition 4: (Vocabulary Container and Case Space, General Definition) We call the pair $\mathbf{VOC} = (D, L)$ the *vocabulary container*. The related *case space* is $C = D \times L$.

Definition 5: (Case Base) A *case base* is a finite set of cases, $\mathbf{CB} = \{c_1, \ldots, c_n\} \subseteq C$.

B. Similarity Measure

The core of the implicit knowledge modeling of the CBR module is the similarity measure. Adapting standard graph similarity measures from [13] to the MultiNet methodology [10], we developed the new 'integrated sub graph/edit similarity' for answer validation in the MultiNet setting as follows.

1) Largest Common Sub Graph Matching Similarity: One important component of the new integrated similarity measure is a graph matching similarity. While isomorphic measures focus on bijective relations, the largest common sub graph approach allows us to define a non binary similarity measure.

Definition 6: (Largest Common Sub Graph [13]) A graph G is the *largest common sub graph* $G = \mathrm{lcsg}(G_1, G_2)$ of two graphs G_1 and G_2 if $G \subseteq G_1$, $G \subseteq G_2$, and there does not exist a graph G' such that $G' \subseteq G_1$, $G' \subseteq G_2$ and $|G'| > |G|$. $|G| = |N| + |E|$ is given by the node and edge count.

A symmetric[3] version of the largest common sub graph similarity measure can be defined as follows [13]:

$$\mathrm{sim}_{\mathrm{lcsg}}(x, y) = f(1 - \tfrac{|\mathrm{lcsg}(x,y)|}{\max\{|X|,|Y|\}})$$

2) Graph Edit Similarity: The other important component of the new integrated similarity measure is a graph edit similarity. The graph editing difference counts and weights the number of transformations necessary to transform one graph into the other. Its computation is NP-complete [14].[4]

$$\delta_{\mathrm{edit}}(x, y) = \min\{\sum_{i=1}^{k} c(e_i) : e_1, \ldots, e_k \text{ transforms } x \text{ to } y\}$$

It is easy to derive the respective similarity measure:

$$\mathrm{sim}_{\mathrm{edit}}(x, y) = 1 - f(\delta_{\mathrm{edit}}(x, y))\,.$$

The cheaper and fewer the operations required to make the two graphs identical, the smaller is the difference and hence the higher the similarity [13].

3) Integrated Sub Graph / Edit Similarity: The similarity measure we use, takes advantage of both the least common sub graph matching similarity and the graph edit similarity. Generally speaking, we measure the similarities of corresponding graph components e.g. attributes, lists of attributes, nodes

[3] $\mathrm{sim}(x, y) = \mathrm{sim}(y, x)$, the symmetric property is an optional but common property of similarity measures. $f : [0, 1] \to [0, 1]$ is a monotonically decreasing mapping with $f(0) = 1$.

[4] Currently we use no heuristic shortcuts since the limited size of the graphs representing questions and single sentences allows a brute-force computation.

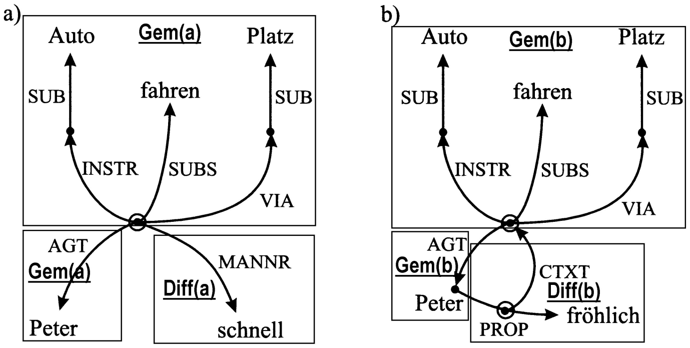

Fig. 1. Illustration of Gem and Diff

and edges with local similarities [12]. If we find pairs with a similarity higher than a threshold we put them into a set of subgraphs and graph components Gem, else in Diff. Then we sum up a certain relevance ω_k for each graph component in Gem and Diff and compute the global similarity from it.

Definition 7: (Global Similarity Measure) For cases X, Y,

$$\mathrm{Sim}_{\mathrm{glob}}(X,Y) = \sum_{n=1}^{m} \omega_n(\mathrm{sim}_n(x_n, y_n)) : x_n \in X, y_n \in Y$$

Definition 8: (Graph Edit Similarity) (i) with $x_i, y_i \subseteq \mathrm{Nodes}(X,Y)$:

$$\mathrm{Sim}_{op}(x_i, y_i) = \mathrm{sim}_{\mathrm{Nodes}}(x_{io}, y_{ip}) \colon o, p = 1, .., \#\mathrm{Nodes\ in\ } x, y$$

(analogously for edges);
(ii) with $x_i, y_i \subseteq \mathrm{Lists}(X,Y)$:

$$\mathrm{sim}_{\mathrm{list}}(x_i, y_i) = \frac{|x_i \cap y_i|}{\max(|x_i|, |y_i|)}$$

(iii) for attributes,

$$\mathrm{sim}_{\mathrm{attribute}}(u, v) = \begin{cases} 1 & : \quad u = v \\ 0 & : \quad \text{else} \end{cases}$$

Let $\mathrm{Gem}, \mathrm{Diff} \subseteq X, Y$ be subgraphs, MultiNet attributes, or components, S_q relevance limits, and ω_k be weights such that

$$\forall q : \mathrm{sim}_q(x_q, y_q) \geq S_q \rightarrow x_q \in \mathrm{Gem}_{x_q}, y_q \in \mathrm{Gem}_{y_q}$$
$$\forall q : \mathrm{sim}_q(x_q, y_q) < S_q \rightarrow x_q \in \mathrm{Diff}_{x_q}, y_q \in \mathrm{Diff}_{y_q}$$

$$|\mathrm{Gem}_{x_q}| = \sum_{k=0}^{\#\text{Elements in } \mathrm{Gem}_{x_q}} \omega_k \,,$$

analogously for Gem_{y_q}, Diff_{x_q}, and Diff_{y_q}. Then

$$\mathrm{Sim}_{\mathrm{items}}(X,Y) = \frac{\frac{|\mathrm{Gem}_{x_q}|}{|\mathrm{Gem}_{x_q}| + |\mathrm{Diff}_{x_q}|} + \frac{|\mathrm{Gem}_{y_q}|}{|\mathrm{Gem}_{y_q}| + |\mathrm{Diff}_{y_q}|}}{2} \,.$$

Fig. 1 illustrates these concepts (example taken from [10], with added Gem and Diff information).

IV. Machine Learning Approach

Answer reranking aims at shifting correct answer candidates for a question to top ranks. Our goal is to improve a solution for answer reranking through CBR, so that the QA system can profit from every new user annotation extending the case base. To this end, the results of the CBR stage are turned into numeric features. A ranking model determined by a supervised learning-to-rank approach combines these CBR-based features with other validation features [1].

A. CBR-Based Input to the Ranker

For each answer candidate, the CBR stage generates the following output data based on the global similarity measure:

- *bestDec* = YES or NO (class of the case with the highest similarity score). *bestDec* expresses that the known answer candidate from the case base was annotated correct (YES) or incorrect (NO).
- *bestSim* = similarity score of the best case that justifies the choice of YES or NO according to *bestDec*.
- *nonBestSim* = similarity score of the best case that does not justify the chosen decision, thus providing evidence for the opposite choice. It always holds that *nonBestSim* ≤ *bestSim*.
- *nonBestRank* = position of the first case in the similarity ranking that does not justify the chosen decision (e.g., best rank of a NO case when the known case with the highest similarity score is annotated YES).

B. CBR-Based Features

Given the CBR result data, several derived attributes can be computed. By means of these CBR features, the CBR result can be combined with the feature set already used by the LogAnswer system for answer reranking.

1) $\textit{cbrSignedSim} = \begin{cases} \textit{bestSim} & : \quad \textit{bestDec} = \text{YES} \\ -\textit{bestSim} & : \quad \text{else} \end{cases}$
2) $\textit{cbrYesSim} = \begin{cases} \textit{bestSim} & : \quad \textit{bestDec} = \text{YES} \\ \textit{nonBestSim} & : \quad \text{else} \end{cases}$
3) $\textit{cbrNoSim} = \begin{cases} \textit{nonBestSim} & : \quad \textit{bestDec} = \text{YES} \\ \textit{bestSim} & : \quad \text{else} \end{cases}$
4) $\textit{cbrSimDiff} = \begin{cases} \textit{bnDiff} & : \quad \textit{bestDec} = \text{YES} \\ -\textit{bnDiff} & : \quad \text{else} \end{cases}$
where *bnDiff* = *bestSim* − *nonBestSim*.
5) $\textit{cbrRelDiff} = \begin{cases} \textit{nbQuot} & : \quad \textit{bestDec} = \text{YES} \\ -\textit{nbQuot} & : \quad \text{else} \end{cases}$
where *nbQuot* = *nonBestSim*/*bestSim*, and we assume that *bestSim* $\neq 0$. Otherwise, the result shall be 0.
6) $\textit{cbrSignedRank} \begin{cases} \textit{nonBestRank} & : \quad \textit{bestDec} = \text{YES} \\ -\textit{nonBestRank} & : \quad \text{else} \end{cases}$

C. Learning-to-Rank Model

Rank-optimizing decision trees [2] are used for learning an answer reranking model from the existing answer validation features of LogAnswer [1] and the new CBR features. This supervised learning method was specifically developed for imbalanced data, such as typical sets of answer candidates that contain only a small fraction of correct answers. The approach can incorporate monotonicity specifications with respect to the attributes, expressing that the ranking of an item should improve with higher values and decrease with smaller values. For the existing feature set, we adopt the monotonicity specifications documented in [1]. All CBR features except *cbrNoSim* are treated as being positively linked to ranking position, while *cbrNoSim* is declared to have a negative effect. The final ML ranker is an ensemble of ten rank-optimizing decision trees, obtained by stratified bagging [2], whose individual probability estimates are combined by averaging. Each tree is pruned to a size of 40 splits. For efficiency, the CBR features are quantized in bins of 0.001. As was shown in [2], bagging of rank-optimizing decision trees outperforms decision-tree learners (and other common ML techniques) in answer selection tasks.

V. Experimental Results

A. Data Set Used for Experiments

The data set used for the experiments was generated by retrieving answer candidates for questions in the QA@CLEF 2007 and QA@CLEF 2008 test sets for German.[5] Since the CLEF 2008 questions involve anaphoric references which are not of interest for our present purposes, these references were resolved manually by filling in the proper antecedents. Due to our focus of CBR on the semantic network level, we only kept questions and answer candidates for which linguistic processing managed to construct a network. Due to the focus on learning to rank, we only kept questions with at least one correct answer candidate (otherwise reranking is pointless).[6] Duplicate answer candidates (identical but from different documents) were removed. This left us with 254 questions and ca. 15,000 items in the data set.

They were manually annotated for correctness as answers to the question.

B. Case Retrieval Experiments

We measured classification accuracy with a case base constructed from the questions and answer candidates as described in the previous section.[7] For the optimization task we used simple hill climbing algorithms. We rejected the optimal A* approach only because of its calculation complexity. We conducted several experiments, namely:

1) Retrieve the same meaning or question/answer candidate pairs involving synonyms. As for every query, there is a very similar correct best match in the full case base, 100% are correctly retrieved.
2) Retrieve known questions with new answer candidates. The system finds the question and classifies correctly for 73% of the cases (56% for correct cases).
3) Retrieve new questions with new answer candidates. Here we temporarily delete the probed question in the case base, so that there is no very close best match. Classification rate drops to 59% (29% for correct answers).
4) Retrieve new questions with new answer candidate pairs with a case base optimized to get optimal results for the classification of correct answer candidates for already known question/answer candidate pairs. A simple hill climbing algorithm was used for deleting incorrect cases disturbing the classification of the correct cases (questions with correct answer candidates). The classification of correct cases increases to 100% (overall: 82%).
5) Retrieve new questions with new answer candidate pairs with a case base optimized to get optimal overall results for already known questions/answer candidate pairs. Here, a simple hill climbing technique served to delete those correct cases that disturb the overall result. The overall result of 87% (96% for correct answers) is our optimum with the current setup for questions not covered by the case base. Obviously the result degrades with less experience available in the case base.
6) Retrieve new questions with new answer candidate pairs with a raw case base, in a 3-fold cross validation test. The setup is like 3), with the case base reduced by one third when testing for each fold. The 56% classification rate on the test folds (23% for correct answers) forms a baseline for comparison with setup 7) and 8).
7) Retrieve new questions with new answer candidate pairs with a case base optimized to get optimal results for the classification of correct answer candidates, in a 3-fold cross validation test. Classification rate is 56% overall, 40% for correct answers. As expected, the effect of the hill climbing optimization is smaller than in 4).
8) Retrieve new questions with new answer candidate pairs with a case base optimized to get optimal overall results, in a 3-fold cross validation. Classification rate is 55%, 29% for correct answers. As in 7), the effect of the optimization reduces.

The results of the experiments are graphically summarized in Fig. 2 (overall percentages) and Fig. 3 (comparison of detailed percentages).

C. Answer Reranking Experiments Based on CBR Features

1) Data Set and Basic Setup for Reranking Experiments: For the learning-to-rank experiments, we needed training data with realistic values of the CBR features. The folds used for optimizing the case base cannot be used once again for training the ranker, since in this case, the CBR features would already have been adjusted to the data. We therefore composed the results of the CBR method for the three test folds (of unseen cases) into a data set for training and testing the ML ranker. The CBR feature values then stem from three different configurations of the case base, which likely makes the ML problem harder, since the ranker cannot adjust to a single CBR model. But the chosen data set spans all original data, and it contains only values of CBR features for previously unseen data. This data set of existing validation features plus CBR

[5] See http://www.clef-initiative.eu/ for more information about CLEF.

[6] Note that due to these restrictions, one cannot extend the experiment to the full QA@CLEF 2007 and 2008 test sets. Our system has fallback techniques to handle the 146 questions that were taken aside, but they are not of interest here since our back-off approach does not involve CBR, yet.

[7] All data sets will be made **publicly available** upon acceptance.

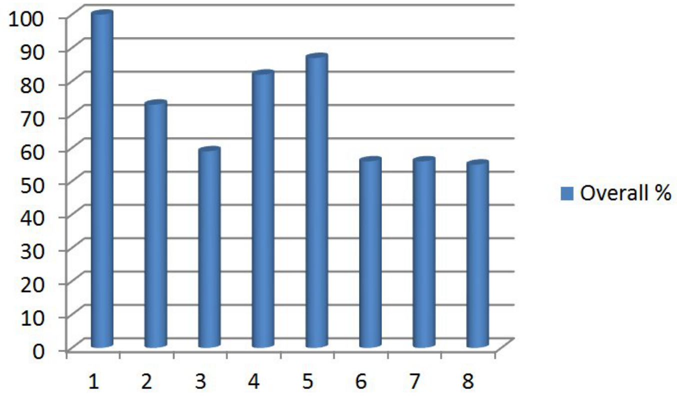

Fig. 2. Overall Percentage Results of Case Retrieval Experiments

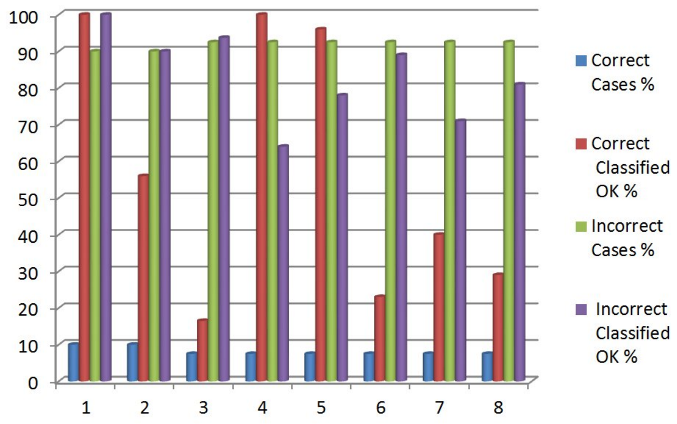

Fig. 3. Detailed Percentage Results in Comparison

results is used in a 10-fold cross validation experiment for evaluating the feature set and the learning-to-rank method.

2) Direct Reranking by Individual CBR Features: The results of the CBR approach alone, using the CBR features directly for ranking, are shown in Table I, using the full non-optimized case base.[8] The best results for each column are shown bold. The metrics used are the mean-reciprocal rank (MRR) and a criterion ANS-i, $i \in \{1, 2, 3, 4, 5\}$ for the proportion of test questions with at least one correct candidate on ranks $1, \ldots, i$. The models *cbrRelDiff*, *cbrSimDiff*, *cbrSignedSim*, *cbrYesSim*, *cbrSignedRank* and *cbrNoSim* use the corresponding CBR attribute directly for ranking. Our baseline is *irScore*, the original ranking of the candidate retrieval system. Despite the value of our CBR setup for classification, the quantities derived from the CBR-based similarity measure perform worse than the baseline in answer ranking.

3) Evaluation of the Learning-to-Rank Approach: The negative picture concerning reranking by isolated CBR features changes as soon as the CBR features are combined with the existing answer validation features of LogAnswer, applying rank-optimizing decision trees for learning suitable rankers. The results of the learning-to-rank approach for various combinations of feature sets are shown in Tables II–IV.

TABLE I
DIRECT CBR FEATURE RANKING FOR NON-OPTIMIZED CASE BASE

Feature	MRR	ANS-1	ANS-2	ANS-3	ANS-4	ANS-5
irScore	**0.48**	**0.33**	**0.46**	**0.55**	**0.62**	**0.67**
cbrSimDiff	0.29	0.17	0.26	0.32	0.39	0.41
cbrRelDiff	0.29	0.17	0.26	0.31	0.39	0.41
cbrSignedSim	0.28	0.16	0.24	0.31	0.37	0.44
cbrYesSim	0.27	0.13	0.20	0.28	0.35	0.43
cbrSignedRank	0.22	0.08	0.18	0.21	0.28	0.33
cbrNoSim	0.15	0.04	0.09	0.15	0.17	0.21

TABLE II
ML RESULTS FOR UNOPTIMIZED CASE BASE

Model	MRR	ANS-1	ANS-2	ANS-3	ANS-4	ANS-5
DSC3	**0.72**	**0.60**	0.74	**0.83**	0.85	0.88
DS3	**0.72**	0.58	**0.76**	**0.83**	**0.87**	0.89
S	0.71	0.58	0.74	0.82	0.86	**0.90**
SC	0.69	0.55	0.70	0.80	0.85	0.88
irScore	0.48	0.33	0.46	0.55	0.62	0.67
CI	0.38	0.21	0.34	0.46	0.53	0.59
C	0.28	0.15	0.25	0.29	0.37	0.41

TABLE III
ML RESULTS FOR CASE BASE OPTIMIZED FOR CORRECT CANDIDATES

Model	MRR	ANS-1	ANS-2	ANS-3	ANS-4	ANS-5
DSC3	**0.74**	**0.61**	**0.76**	**0.83**	**0.88**	0.89
DS3	0.72	0.58	**0.76**	**0.83**	0.87	0.89
S	0.71	0.58	0.74	0.82	0.86	**0.90**
SC	0.69	0.55	0.72	0.80	0.85	0.86
irScore	0.48	0.33	0.46	0.55	0.62	0.67
CI	0.37	0.22	0.34	0.41	0.47	0.51
C	0.27	0.14	0.24	0.30	0.34	0.37

The feature sets used for training the ML models were D: deep features (based on results of logic-based answer validation, as described in [1]), I: the *irScore* feature, i.e. the original retrieval score of the candidate retrieval system, S: shallow features including I (described in [1]), and C: CBR features. The number 3 is the limit on relaxation steps.

As shown by Table II for the unoptimized case base, combining all existing answer validation features with the new CBR-based features yields the best MRR and the best top-ranked result (ANS-1). As shown by Table III, the same holds when training the ML ranker on a case base optimized for perfect treatment of correct answers. This is our best overall result, with an MRR of 0.74 and a correct top-ranked answer in 61% of the cases. Table IV shows a slight decrease in ranking quality when optimizing the case base for overall classification rate. A possible explanation is the strong imbalance of the data, so that optimizing for classification rate may be detrimental to the treatment of the few correct answer candidates in the case base. As shown by the superior DSC3 result in Table III, it is better to optimize the case base for retrieving correct answers.

[8] The results for the case base optimized for perfect treatment of correct candidates, and for the case based optimized for overall performance, are very similar, so we omit them here.

TABLE IV
ML RESULTS FOR CASE BASE OPTIMIZED FOR OVERALL PERFORMANCE

Model	MRR	ANS-1	ANS-2	ANS-3	ANS-4	ANS-5
DS3	**0.72**	**0.58**	**0.76**	**0.83**	**0.87**	0.89
DSC3	0.71	**0.58**	0.74	0.82	0.86	0.88
S	0.71	**0.58**	0.74	0.82	0.86	**0.90**
SC	0.70	0.56	0.73	0.81	0.84	0.88
irScore	0.48	0.33	0.46	0.55	0.62	0.67
CI	0.34	0.18	0.28	0.38	0.46	0.52
C	0.27	0.14	0.22	0.28	0.33	0.41

We considered the usage of CBR features in the best ML ranker DSC3 shown in Table III, by inspecting all branching conditions in the generated trees (since 10 bags of 10 trees each were generated in the cross-validation runs, there are 100 trees to base results on). It turns out that 10.7% of all branching conditions involve *cbrSignedSim* (rank 2 among all 26 features seen by the ML ranker), 10.5% of all splits involve *cbrRelDiff* (rank 3), 7.1% *cbrSignedRank* (rank 5), 5.6% *cbrNoSim* (rank 8), 4.6% *cbrYesSim* (rank 10), and 4.0% of all branching conditions involve *cbrSimDiff* (rank 12). In total, 42.5% of all split conditions in the learned trees involve one of the CBR attributes. This demonstrates a strong impact of CBR results on answer scores, so that new experience in the case base will also become effective in answer reranking.

VI. CONCLUSIONS AND FUTURE WORK

We have presented an integrated CBR approach to answer validation and reranking for QA systems. Since each annotated answer candidate for a question provides evidence for validating answer candidates for future questions, this approach promises a continuous increase in answer quality, given that the QA system allows user feedback extending the case base. By comparing the semantic structure of questions and potential answers, the proposed CBR approach with its graph similarity measure complements the logic-based answer validation that is already implemented in LogAnswer. In the paper, we have emphasized the use of CBR techniques and their integration into a working learning-to-rank approach through CBR-related features. Data from QA@CLEF served to evaluate case retrieval and the achieved reranking quality. As to classification rate (percentage of best retrieved cases with proper class label), the CBR approach works extremely well for question/answer candidate pairs of the same meaning, and quite good if the question was already asked and correctly answered, even if the correct answer candidate is of a different semantic construction. The less similar the questions and answer candidates become to known ones, the worse the quality of the best matches. Concerning the integration of CBR results into the existing answer reranking solution, experiments show that the best learned models include CBR features, achieving an MRR up to 0.74 with a correct top-ranked answer shown in 61% of the cases. The advantage over the original approach without CBR will increase with time due to automatic improvements with new user annotations. The massive use of CBR features by the learning- to-rank method, comprising 42.5% of all branching conditions in the learned decision trees, shows that the CBR features have a strong impact on answer reranking - as needed for improvements in the case base to push ranking results. For the future, one of our goals is to demonstrate this positive effect of the human-in-the-loop experimentally, simulating user feedback by feeding the case base with growing sets of questions/annotations.

Since at present, the isolated CBR features perform worse than the *irScore* baseline of the candidate retrieval system, or main focus is on improving the similarity measure for MultiNet graphs. Firstly, we will analyze in depth the local similarity results for finding those local similarity measures that classify best. We will optimize the structure and weights of the global similarity measures for the hard case of question/answer candidate pairs completely different from the case base. To this end, we will also analyze in depth the most similar components of the MultiNet graphs of correct question/answer pairs to find generalizations and abstractions that will further help to classify those question/answer candidate pairs that do not share the specific lexical concepts of cases in the case base, but rather show parallels in the MultiNet graph structure. We aim to optimize the similarity measure by using more adapted, relevant implicit knowledge and integrate more explicit knowledge [12]. Generalized and abstracted cases [13] promise a more general view in elaborating the similarities.

REFERENCES

[1] U. Furbach, I. Glöckner, and B. Pelzer, “An application of automated reasoning in natural language question answering,” *AI Communications*, vol. 23, no. 2-3, pp. 241–265, 2010.

[2] I. Glöckner, “Finding answer passages with rank optimizing decision trees,” in *Proc. of the 8th Int. Conference on Machine Learning and Applications (ICMLA-09)*. IEEE Press, 2009, pp. 208–214.

[3] R. D. Burke, K. J. Hammond, V. A. Kulyukin, S. L. Lytinen, N. Tomuro, and S. Schoenberg, “Question answering from frequently-asked question files: Experiences with the FAQ Finder system,” The University of Chicago, Computer Science Department, Chicago, TR 97-05, Jun. 1997.

[4] K. Jia, X. Pang, and Z. Li, “Question answering system in network education based on FAQ,” in *Proceedings ICYCS ’08*. IEEE Computer Society, 2008, pp. 2577–2581.

[5] M. Lenz, A. Hübner, and M. Kunze, “Question answering with textual CBR,” in *Proceedings of the 3rd Int. Conf. on Flexible Query Answering Systems (FQAS ’98)*. London, UK: Springer, 1998, pp. 236–247.

[6] M. Lenz and K. Ashley, Eds., *Proceedings of the AAAI98 Workshop on Textual Case-Based Reasoning*. AAAI Press, 1998.

[7] G. Fouque, W. W. Chu, and H. Yau, “A case-based reasoning approach for associative query answering,” in *Proc. of Methodologies for Int. Systems 8th Int. Symposium (ISMIS’94*, 1994, pp. 183–192.

[8] I. Glöckner and B. Pelzer, “The LogAnswer project at CLEF 2009,” in *Results of the CLEF 2009 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2009 Workshop*, Sep. 2009.

[9] S. Hartrumpf, *Hybrid Disambiguation in Natural Language Analysis*. Osnabrück, Germany: Der Andere Verlag, 2003.

[10] H. Helbig, *Knowledge Representation and the Semantics of Natural Language*. Berlin: Springer, 2006.

[11] A. Aamodt, “Knowledge-intensive, integrated approach to problem solving and sustained learning,” Ph.D. dissertation, University of Trondheim, Department of Computer Science and Telematics, 1991.

[12] K.-H. Weis, “Evaluation fallbasierter Systeme – evaluation of case based reasoning systems,” Master’s thesis, University Kaiserslautern, 1995.

[13] R. Bergmann, *Experience Management – Foundations, Development Methodology and Internet-Based Applications*. Berlin: Springer, 2002.

[14] H. Bunke and B. T. Messmer, “Similarity measures for structured representations,” in *Topics in Case-Based Reasoning: First European Workshop, EWCBR-93, selected papers*, S. Wess, K.-D. Althoff, and M. M. Richter, Eds. Berlin: Springer, 1994, pp. 106–118.