Chapter 8
Component-Based Evaluation for Question Answering

Teruko Mitamura and Eric Nyberg

Abstract This chapter describes the component-based evaluation of automatic question answering (QA) systems, which was pioneered in the NTCIR-7 ACLIA challenge and has became a fundamental part of QA system development, especially for difficult real-world datasets which require a multi-strategy, multi-component approach. We summarize the history of component evaluation for QA and describe more recent work at Carnegie Mellon (on TREC Genomics, BioASQ, and LiveQA datasets) which has descended directly from our experiences in NTCIR.

8.1 Introduction

In this chapter, we first describe the component-based evaluations for question answering that were developed as part of past NTCIR challenges. We introduce the CMU JAVELIN Cross-lingual Question Answering (CLQA) system and show how the JAVELIN architecture supports component-level evaluation, which can accelerate overall system development. This component-based evaluation concept was used in the NTCIR-7 ACLIA tasks, not only to evaluate each component but also to evaluate different combinations of Information Retrieval (IR) and Question Answering (QA) modules.

In later sections, we describe more recent developments in component-based evaluation within the Open Advancement of Question Answering (OAQA) and Configuration Space Exploration (CSE) projects. We also describe automatic component evaluation for biomedical QA systems. All of these later developments were influenced by the original vision of component-based evaluation embodied in the NTCIR QA tasks. To conclude, we discuss remaining challenges and future directions for component-based evaluation in QA.

T. Mitamura (✉) · E. Nyberg
Carnegie Mellon University, Pittsburgh, PA, USA
e-mail: teruko@andrew.cmu.edu

E. Nyberg
e-mail: ehn@cs.cmu.edu

T. Sakai et al. (eds.), *Evaluating Information Retrieval and Access Tasks*,
The Information Retrieval Series 43,
https://doi.org/10.1007/978-981-15-5554-1_8

8.1.1 History of Component-Based Evaluation in QA

The JAVELIN Cross Language Question Answering (CLQA) system, developed by the Language Technologies Instutute (LTI) at Carnegie Mellon University (CMU) had five main components: question analysis, keyword translation, document retrieval, information extraction, and answer generation (Mitamura et al. 2007). This system contains an English-to-Japanese QA system and an English-to-Chinese QA system with the same overall architecture, which supported direct comparison of the two systems on a per-module basis. After analyzing the observed performance of each module on the evaluation data, we created gold-standard data (perfect input) for each module in order to determine upper bounds on module performance. The overall architecture is shown in Fig. 8.1.

The Question Analysis (QA) module is responsible for parsing the input question, choosing the appropriate answer type, and producing a set of keywords. The Translation Module (TM) translates the keywords into task-specific languages. The Retrieval Strategist (RS) module is responsible for finding relevant documents which might contain answers to the question, using translated keywords produced by the Translation Module. The Information Extractor (IX) module extracts answers from the relevant documents. The Answer Generation (AG) module normalizes the answers and ranks them in order of correctness.

Although traditional QA systems consist of several modules with a cascaded approach, as far as we know the JAVELIN CLQA system was the first one to incorporate component-based evaluation for QA. We participated in the NTCIR-5 CLQA1 task and demonstrated our results (Lin et al. 2005). A more detailed analysis of our component-based evaluation was presented at LREC 2006 (Shima et al. 2006).

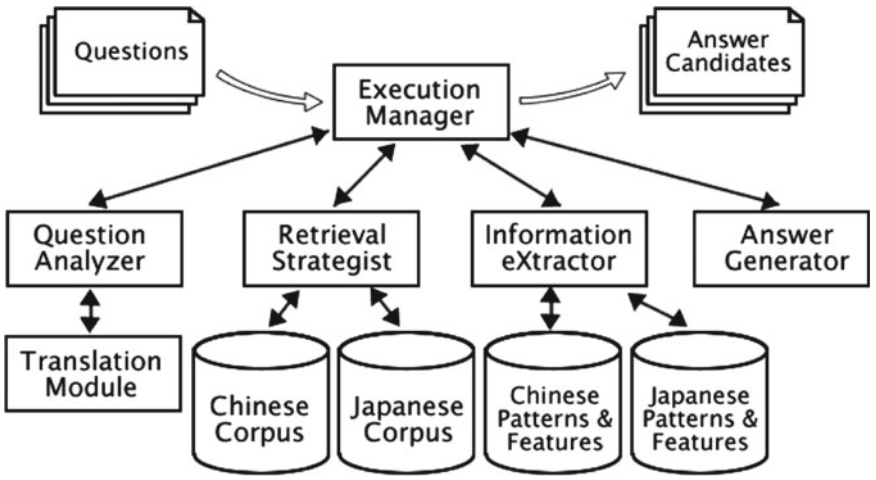

Fig. 8.1 JAVELIN architecture

8.1.2 Contributions of NTCIR

NTCIR first included a question answering challenge (QAC) evaluation for Japanese in 2002 (NTCIR-3). The NTCIR-4 and the NTCIR-5 challenges continued to include QAC tasks in 2004 and 2005 respectively. The NTCIR-5 challenge also added the first cross-lingual QA task, which contained five subtasks for three languages: English, Japanese, and Chinese. The JAVELIN system was evaluated on the CLQA tasks for all three languages. When developing cross-lingual capabilities with three languages, system and component development became more complicated, and error analysis became very challenging. Therefore, we developed a component-based evaluation approach for error analysis and improvement of the JAVELIN CLQA system (Lin et al. 2005; Shima et al. 2006).

Input questions in English are processed by these modules in the order listed above. The answer candidates are returned in one of the two target languages (Japanese and Chinese) as final outputs. The QA module is responsible for parsing the input question, choosing the expected answer type, and producing a set of keywords. The QA module calls the Translation Module, which translates the keywords into the language(s) required by the task.

In order to gain different perspectives on the tasks and our system's performance, a module-by-module analysis was performed. We used the formal run dataset from NTCIR task CLQA1, which includes English–Chinese (EC) and English–Japanese (EJ) subtasks. 200 input questions were provided for each of the subtasks. This analysis was based on gold-standard answer data, which also provides information about the documents that contain the correct answer for each question. We judged the QA module by the accuracy of its answer type classification, and the Translation Module by the accuracy of its keyword translation. For the RS and IX modules, if a correct document or answer is returned, regardless of its ranking, we consider the module to be successful. To separate the effects of errors introduced by earlier modules, we created gold-standard data by manually correcting answer type and keyword translation errors. We also create "perfect" IX input using the gold-standard document set. In Table 8.1, the overall performance (top 1 average accuracy) is shown in the last two columns of the top rows for EC and EJ. The symbol "R" indicates recall versus the standard gold answer set; the symbol "R+U" indicates recall versus the standard gold answer set plus other (unofficial) correct answers ("Unsupported"). If we examine only such global measures, we will not be able to understand the performance of individual modules in a complex system.

Our analysis of per-module performance from gold-standard input shows that the QA module and the RS module are already performing fairly well, but there is still room in the IX module and the AG module for future improvement.

Table 8.1 Modular performance analysis (Shima et al. 2006)

	Gold standard input	AType accuracy (%)	TM accuracy (%)	RS top 15 (%)	IX top 100 (%)	MRR	Overall top 1 R (%)	Top 1 R+U (%)
EC	None	86.5	69.3	30.5	30.0	0.130	7.5	9.5
EC	TM	86.5	–	57.5	50.0	0.254	9.5	20.0
EC	TM+AType	–	–	57.5	50.5	0.260	9.5	20.5
EC	TM+AType+RS	–	–	–	63.0	0.489	41.0	43.0
EJ	None	93.5	72.6	44.5	31.5	0.116	10.0	12.5
EJ	TM	93.5	–	67.0	41.5	0.154	9.5	15.0
EJ	TM+AType	–	–	68.0	45.0	0.164	10.0	15.5
EJ	TM+AType+RS	–	–	–	51.5	0.381	32.0	32.5

8.2 Component-Based Evaluation in NTCIR

In 2007, LTI/CMU became an organizer of Advanced Cross-lingual Information Access (ACLIA) task for NTCIR-7. In this task, we started the formal component-based evaluation for Japanese (JA), Simplified Chinese (CS), Traditional Chinese (CT), and English for the first time (Mitamura et al. 2008). There were two major tasks: (1) Information Retrieval for Question Answering (IR4QA) and (2) Complex Cross-Lingual Question Answering (CCLQA) tasks. Within the CCLQA task, we had three subtasks: Question Analysis track, CCLQA Main Track, and IR4QA+CCLQA collaboration tracks (obligatory track and optional track). The ACLIA task data flow is illustrated in Fig. 8.2.

As a central problem in question answering evaluation, the lack of standardization made it difficult to compare systems under a shared condition. In NLP research at that time, system design was moving away from monolithic, black-box architectures and more toward modular, architectural approaches that include an algorithm-independent formulation of the system's data structures and data flows, so that multiple algorithms implementing a particular function can be evaluated on the same task. Therefore, the ACLIA data flow includes a pre-defined schema for representing the inputs and outputs of the document retrieval step, as illustrated in Fig. 8.2. This novel standardization effort made it possible to evaluate IR4QA (Information Retrieval for Question Answering) in the context of a closely related QA task. During the evaluation, the question text and QA system question analysis results were provided as input to the IR4QA task, which produced retrieval results that were subsequently fed back into the end-to-end QA systems. The modular design and XML interchange format supported by the ACLIA architecture made it possible to perform such embedded evaluations in a straightforward manner.

The modular design of this evaluation data flow is motivated by the following goals: (a) to make it possible for participants to contribute component algorithms to an evaluation, even if they cannot field an end-to-end system; (b) to make it possible to conduct evaluations on a per-module basis, in order to target metrics and error

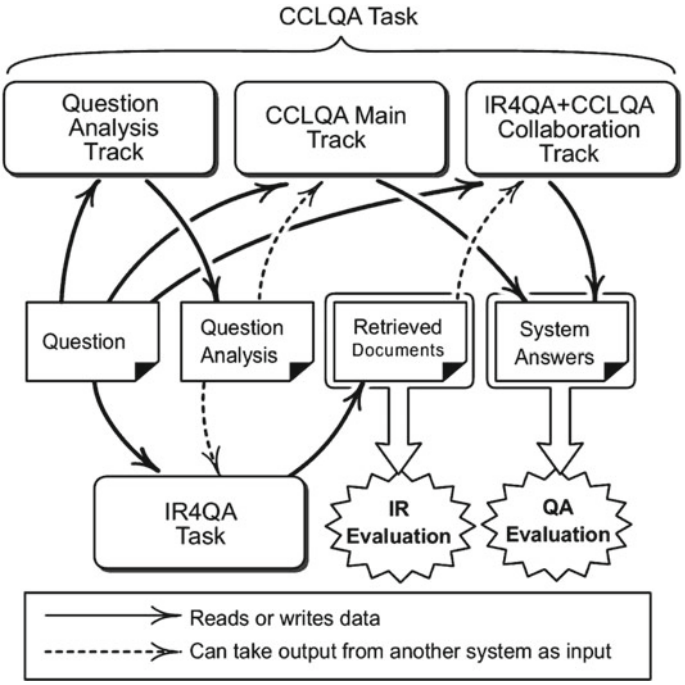

Fig. 8.2 Data flow in ACLIA task cluster showing how interchangeable data model made inter-system and inter-task collaboration possible (Mitamura et al. 2008)

analysis on important bottlenecks in the end-to-end system; and (c) to determine which combination of algorithms works best by combining the results from various modules built by different participants.

8.2.1 Shared Data Schema and Tracks

In order to combine a Cross-Lingual Information Retrieval (CLIR) module with a cross-lingual Question Answering (CLQA) system for module-based evaluation, we defined five types of XML schema to support exchange of results among participants and submission of results to be evaluated:

- **Topic format**: The organizer distributes topics in this format for formal run input to IR4QA and CCLQA systems.
- **Question Analysis format**: CCLQA participants who chose to share Question Analysis results submit their data in this format. IR4QA participants can accept task input in this format.
- **IR4QA submission format**: IR4QA participants submit results in this format.

- **CCLQA submission format:** CCLQA participants submit results in this format.
- **Gold-Standard Format**: Organizer distributes CCLQA gold-standard data in this format.

Participants in the ACLIA CCLQA task submitted results for the following four tracks:

- **Question Analysis Track**: Question Analysis results contain key terms and answer types extracted from the input question. These data are submitted by CCLQA participants and released to IR4QA participants.
- **CCLQA Main Track**: For each topic, a system returned a list of system responses (i.e., answers to the question), and human assessors evaluated them. Participants submitted a maximum of three runs for each language pair.
- **IR4QA+CCLQA Collaboration Track (obligatory)**: Using possibly relevant documents retrieved by the IR4QA participants, a CCLQA system-generated QA results in the same format used in the main track. Since we encouraged participants to compare multiple IR4QA results, we did not restrict the maximum number of collaboration runs submitted and used automatic measures to evaluate the results. In the obligatory collaboration track, only the top 50 documents returned by each IR4QA system for each question were utilized.
- **IR4QA+CCLQA Collaboration Track (optional)**: This collaboration track was identical to the obligatory collaboration track, except that participants were able to use the full list of IR4QA results available for each question (up to 1000 documents per-topic).

8.2.2 Shared Evaluation Metrics and Process

In order to build an answer key for evaluation, third party assessors created a set of weighted nuggets for each topic. A "nugget" is defined as the minimum unit of correct information that satisfies the information need.

In this section, we present the evaluation framework used in ACLIA, which is based on weighted nuggets. Both human-in-the-loop evaluation and automatic evaluation were conducted using the same topics and metrics. The primary difference is in the step where nuggets in system responses are matched with gold-standard nuggets. During human assessment, this step is performed manually by human assessors, who judge whether each system response nugget matches a gold-standard nugget. In automatic evaluation, this decision is made automatically. The subsections that follow, we detail the differences between these two types of evaluation.

8.2.2.1 Human-in-the-loop Evaluation Metrics

In CCLQA, we evaluate how well a QA system can return answers that satisfy information needs on average, given a set of natural language questions. We adopted the

nugget pyramid evaluation method (Lin and Demner-Fushman 2006) for evaluating CCLQA results, which requires only that human assessors make a binary decision whether a system response matches a gold-standard "vital" nugget (necessary for the answer to be correct) or "ok" nugget (not necessary, but not incorrect). This method was used in the TREC 2005 QA track for evaluating definition questions, and in the TREC 2006–2007 QA tracks for evaluating "other" questions. We evaluated each submitted run by calculating the macroaverage F-score over all questions in the formal run dataset.

In the TREC evaluations, a character allowance parameter C is set to 100 non-whitespace characters for English (Voorhees 2003). Based on the micro-average character length of the nuggets in the formal run dataset, we derived settings of C = 18 for CS, C = 27 for CT and C = 24 for JA.

Note that precision is an approximation, imposing a simple length penalty on the System Response (SR). This is due to Voorhees' observation that "nugget precision is much more difficult to compute since there is no effective way of enumerating all the concepts in a response" (Voorhees 2004). The precision is a length-based approximation with a value of 1 as long as the total system response length per question is less than the allowance, i.e., C times the number of nuggets defined for a topic. If the total length exceeds the allowance, the score is penalized. Therefore, although there is no limit on the number of SRs submitted for a question, a long list of SRs harms the final F-score.

The F ($\beta = 3$) or simply F3 score has emphasizes recall over precision, with the β value of 3 indicating that recall is weighted three times as much as precision. Historically, a β of 5 was suggested by a pilot study on definitional QA evaluation (Voorhees 2003). In the later TREC QA tasks, the value has been to 3.

8.2.2.2 Automatic Evaluation Metrics

ACLIA also utilized automatic evaluation metrics for evaluating the large number of IR4QA+CCLQA Collaboration track runs. Automatic evaluation is also useful during developing, where it provides rapid feedback on algorithmic variations under test. The main goal of research in automatic evaluation is to devise an automatic metric for scoring that correlates well with human judgment. The key technical requirement for automatic evaluation of complex QA is a real-valued matching function that provides a high score to system responses that match a gold-standard answer nugget, with a high degree of correlation with human judgments on the same task.

The simplest nugget matching procedure is exact match of the nugget text within the text of the system response. Although exact string match (or matching with simple regular expressions) works well for automatic evaluation of factoid QA, this model does not work well for complex QA, since nuggets are not exact texts extracted from the corpus text; the matching between nuggets and system responses requires a degree of understanding that cannot be approximated by a string or regular expression match for all acceptable system responses, even for a single corpus.

$$a_{BINARIZED} = \sum_{n \in Nuggets} \max_{s \in SRs} \mathrm{I}_{\theta}(n, s)$$

$$\mathrm{I}_{\theta}(n, s) = \begin{cases} 1 : \mathrm{NuggetRecall}_{\mathrm{token}}(n, s) > \theta \\ 0 : \mathrm{otherwise} \end{cases}$$

Fig. 8.3 Formulas of the binarized metric used for official ACLIA automatic evaluation (Mitamura et al. 2008)

For the evaluation of complex questions in the TREC QA track, Lin and Demner-Fushman (2006) devised an automatic evaluation metric called POURPRE. Since the TREC target language was English, the evaluation procedure simply tokenized answer texts into individual words as the smallest units of meaning for token matching. In contrast, the ACLIA evaluation metric tokenized Japanese and Chinese texts into character unigrams. We did not extract word-based unigrams since automatic segmentation of CS, CT, and JA texts is non-trivial; these languages lack white space and there are no general rules for comprehensive word segmentation. Since a single character in these languages can bear a distinct unit of meaning, we chose to segment texts into character unigrams, a strategy that has been followed for other NLP tasks in Asian languages (e.g., Named Entity Recognition Asahara and Matsumoto 2003). One of the disadvantages of POURPRE is that it gives a partial score to a system response if it has at least one common token with any one of the nuggets. To avoid over-estimating the score via aggregation of many such partial scores, we devised a novel metric by mapping the POURPRE soft match score values into binary values (see Fig. 8.3). We set the threshold θ to be somewhere in between no match and an exact match, i.e., 0.5, and we used this BINARIZED metric as our official automatic evaluation metric for ACLIA.

Reliability of Automatic Evaluation: We compared per-run (# of data points = # of human evaluated runs for all languages) and per-topic (# of data points = # of human evaluated runs for all languages times # of topics) correlation between scores from human-in-the-loop evaluation and automatic evaluation. The following Table 8.2 from the ACLIA Overview (Mitamura et al. 2008) shows that the correlation between the automatic and human evaluation metrics.

The Pearson measure indicates the correlation between individual scores, while the Kendall measure indicates the rank correlation between sets of data points. The results show that our novel nugget matching algorithm BINARIZED outperformed SOFTMATCH for both correlation measures, and we chose BINARIZED as the official automatic evaluation metric for the CCLQA task.

Table 8.2 Per-run and per-topic correlation between automatic nugget matching and human judgment (Mitamura et al. 2008)

Algorithm	Token	Per-run (N = 40) Pearson	Per-run (N = 40) Kendall	Per-topic (N = 40 × 100) Pearson	Per-topic (N = 40 × 100) Kendall
Exactmatch	Char	0.4490	0.2364	0.5272	0.4054
Softmatch	Char	0.6300	0.3479	0.6383	0.4230
Binarized	Char	0.7382	0.4506	0.6758	0.5228

8.3 Recent Developments in Component Evaluation

The introduction of modular QA design and component-based QA evaluation by NTCIR had a strong influence on subsequent research in applied QA systems. In this section, we summarize key developments in QA research that followed directly from our experiences with NTCIR.

8.3.1 Open Advancement of Question Answering

Shared modular APIs and common data exchange formats have become fundamental requirements for general language processing frameworks like UIMA (Ferrucci et al. 2009a) and specific language applications (like the Jeopardy! Challenge) (Ferrucci et al. 2010). In 2009, a group of academic and industry researchers published a technical report on the fundamental requirements for the Open Advancement of Question Answering (OAQA) (Ferrucci et al. 2009b); chief among these requirements are the shared modular design, common data formats, and automatic evaluation metrics first introduced by NTCIR:

> To support this vision of shared modules, dataflows, and evaluation measures, an open collaboration will include a shared logical architecture—a formal API definition for the processing modules in the QA system, and the data objects passed between them. For any given configuration of components, standardized metrics can be applied to the outputs of each module and the end-to-end system to automatically capture system performance at the micro and macro level for each test or evaluation. (Ferrucci et al. 2009b)

> By designing and building a shared infrastructure for system integration and evaluation, we can reduce the cost of interoperation and accelerate the pace of innovation. A shared logical architecture also reduces the overall cost to deploy distributed parallel computing models to reduce research cycle time and improve run-time response. (Ferrucci et al. 2009b)

A group of eight universities followed these principles in collaborating with IBM Research to develop the Watson system for the Jeopardy! challenge (Andrews 2011). The Watson system utilized a shared, modular architecture which allowed the exploration of many different implementations of question-answering components. In

particular, hundreds of components were evaluated, as part of an answer-scoring ensemble that was used to select Watson's final answer for each clue (Ferrucci et al. 2010).

Following the success of the Watson system in the Jeopardy! Challenge (where the system won a tournament against two human champions, Ken Jennings and Brad Rutter), Carnegie Mellon continued to refine the OAQA approach and engaged with other industrial sponsors (most notably, Hoffman-Laroche) to develop open-source architectures and solutions for question answering (discussed below).

8.3.2 Configuration Space Exploration (CSE)

In January of 2012, Carnegie Mellon launched a new project on biomedical question answering, with support from Hoffman-Laroche. Given the goal of building a state-of-the-art QA system for a current dataset (at that time, the TREC Genomics dataset), the CMU team chose to survey and evaluate published approaches (at the level of architecture and modules) to determine the best baseline solution. This triggered a new emphasis on defining and exploring a space of possible end-to-end pipelines and module combinations, rather than selecting and optimizing a single architecture based on preference, convenience, etc. The Configuration Space Exploration project (Garduño et al. 2013) explored the following research questions (taken from Yang et al. 2013):

- How can we formally define a configuration space to capture the various ways of configuring resources, components, and parameter values to produce a working solution? Can we give a formal characterization of the problem of finding an optimal configuration from a given configuration space?
- Is it possible to develop task-independent open-source software that can easily create a standard task framework and incorporate existing tools and efficiently explore a configuration space using distributed computing?
- Given a real-world information processing task, e.g., biomedical question answering, and a set of available resources, algorithms, and toolkits, is it possible to write a descriptor for the configuration space, and then find an optimal configuration in that space using the CSE framework?

The CSE concept of operations is shown in Fig. 8.4. Given a labeled set of input–output pairs (the *information processing task*), the system searches a space of possible solutions (algorithms, toolkits, knowledge bases, etc.) using a set of standard benchmarks (metrics) to determine which solution(s) have the best performance over all the inputs in the task. The goal of CSE is to find an optimal or near-optimal solution while exploring (formally evaluating) only a smart part of the total configuration space.

Based on a shared component architecture and implemented in UIMA, the Configuration Space Exploration (CSE) project was the first to automatically choose an optimal configuration from a set of QA modules and associated parameter values,

given a set of labeled training instances (Garduño et al. 2013). As part of his Ph.D. thesis at Carnegie Mellon, Zi Yang applied the CSE framework to several biomedical information processing problems (Yang 2017). In the following subsection, we discuss the main results of component evaluation for biomedical QA systems.

8.3.3 Component Evaluation for Biomedical QA

Using the Configuration Space Exploration techniques described in the previous subsection (Garduño et al. 2013), a group of researchers at CMU were able to automatically identify a system configuration which signficantly outperformed published baselines for the TREC Genomics task (Yang et al. 2013). Subsequent work showed that it was possible to build high-performance QA systems by applying this optimization approach to an ensemble of subsystems, for the related set of tasks in the BioASQ challenge (Yang et al. 2015).

Table 8.3 shows a summary of the different components that were evaluated for the TREC genomics task: various tokenizers, part-of-speech taggers, named entity recognizers, biomedical knowledge bases, retrieval tools, and reranking algorithms. As shown in Fig. 8.4, the team evaluated about 2,700 different end-to-end configurations, executing over 190 K test examples in order to select the best-performing configuration (Table 8.4). After 24 hours of clock time, the system (running on 30 compute nodes) was able to find a configuration that significantly outperformed the published state of the art on the 2006 TREC Genomics task, achieving a document MAP of 0.56 (versus a published best of 0.54) and a passage MAP of 0.18 (versus a published best of 0.15). Table 8.5 shows the analogous results for the 2007 TREC

Table 8.3 Summary of components integrated for TREC Genomics. (Yang et al. 2013)

Category	Components
NLP tools	LingPipe HMM-based tokenizer
	LingPipe HMM-based POS tagger
	LingPipe HMM-based named entity recognizer
	Rule-based lexical variant generator
KBs	UMLS for syn/acronym expansion
	EntrezGene for syn/acronym expansion
	MeSH for syn/acronym expansion
Retrieval tools	Indri system
Reranking algorithms	Important sentence identification
	Term proximity-based ranking
	Score combination of different retrieval units
	Overlapping passage resolution

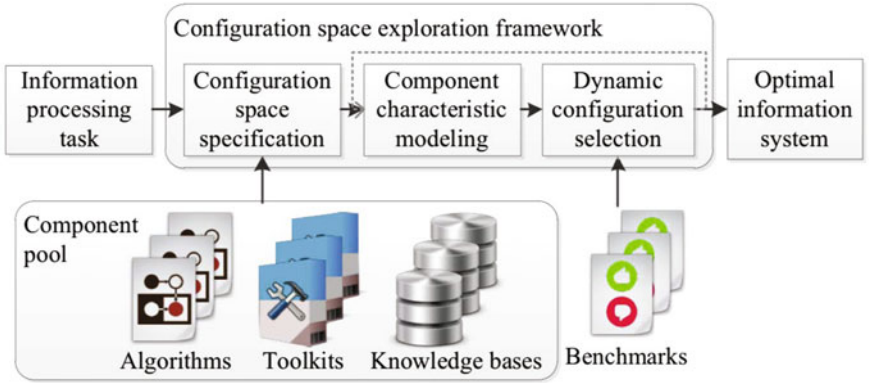

Fig. 8.4 Overview of configuration space exploration framework architecture (Yang et al. 2013)

Table 8.4 Performance of automatically configured components (CSE) versus TREC Genomics 2006 participants (Yang et al. 2013)

	TREC 2006	CSE
No. components	1,000	12
No. configurations	1,000	32
No. traces	92	2,700
No. executions	1,000	190,680
Capacity (hours)	N/A	24
DocMAP max	0.5439	0.5648
DocMAP median	0.3083	0.4770
DocMAP min	0.0198	0.1087
PsgMAP max	0.1486	0.1773
PsgMAP median	0.0345	0.1603
PsgMAP min	0.0007	0.0311

Table 8.5 Performance of automatically configured components versus TREC Genomics 2007 participants (Yang et al. 2013)

	TREC 2007	CSE
DocMAP max	0.3286	0.3144
DocMAP median	0.1897	0.2480
DocMAP min	0.0329	0.2067
PsgMAP max	0.0976	0.0984
PsgMAP median	0.0565	0.0763
PsgMAP min	0.0029	0.0412

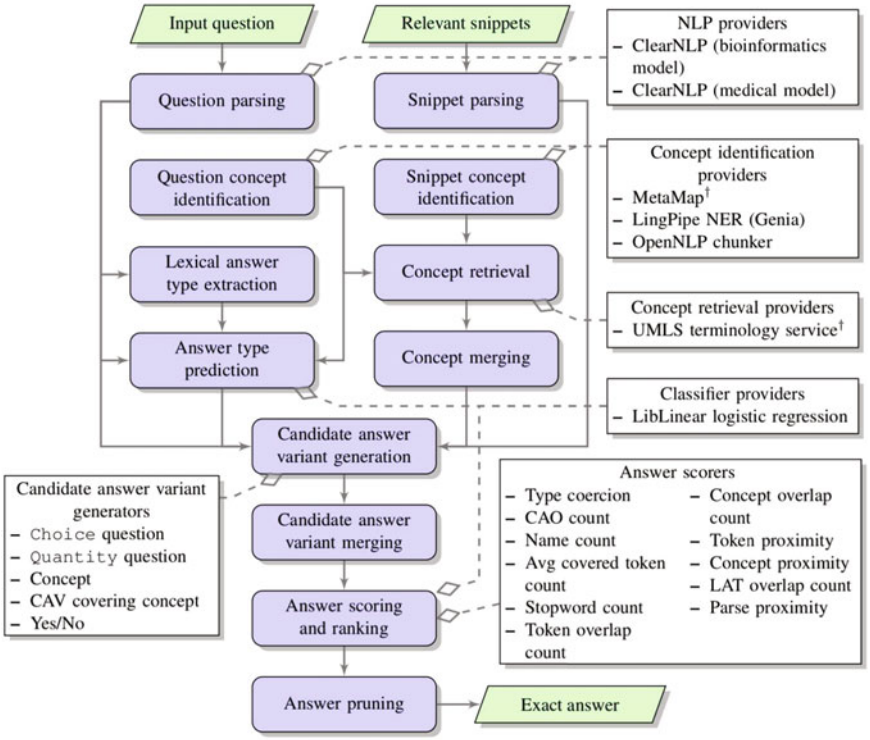

Fig. 8.5 Modular architecture and components for BioASQ phase B (Yang et al. 2015)

Genomics Task, where CSE was also able to find a significantly better combination of components.

The positive results from applying CSE to the TREC Genomics tasks were extended by applying CSE to a much larger, more complex task with many subtasks: The BioASQ Challenge (Chandu et al. 2017; Yang et al. 2015, 2016). Using a shared corpus of biomedical documents (PubMed articles), the BioASQ organizers created a set of interrelated tasks for question answering: retrieval of relevant medical concepts, articles, snippets and RDF triples, plus generation of both exact and "ideal" (summary) answers for each question. Figure 8.5 illustrates the modular architecture used to generate exact answers for 2015 BioASQ Phase B (Yang et al. 2015). Across the five batch tests in Phase B, the CMU system achieved top scores in concept retrieval, snippet retrieval, and exact answer generation. As shown in Fig. 8.5, this involved evaluating and optimizing ensembles of language models, named entity extractors, concept retrievers, classifiers, candidate answer generators, and answer scorers.

8.4 Remaining Challenges and Future Directions

Much recent work in question-answering has focused on neural models which are trained on large numbers of question-answer pairs created by human curators (e.g., SQUAD (Rajpurkar et al. 2016), SQUAD 2 (Rajpurkar et al. 2018). While neural QA approaches are effective when large numbers of labeled training examples are available (e.g., more than 100,000 examples), in practice neural approaches are very sensitive to the distribution of answer texts and corresponding questions that are created by the human curators. For example, a recent study showed that an advanced question curation strategy, using the original answer texts from SQUAD produced a dataset (ParallelQA) that was much tougher for neural models; models evaluated on SQUAD and ParallelQA did approximately 20% worse on ParallelQA (Wadhwa et al. 2018c). In the future, we believe that QA research must focus more energy on defining effective curation strategies, so that the best components and models may be chosen and built into an effective solution using the least amount of labeled data and human resources. In preliminary work, we have adopted a comparative evaluation framework (Wadhwa et al. 2018a) that allows us to compare the performance of different neural QA approaches across datasets, in order to identify the approach with the most general capability.

It is also the case that neural approaches to QA often assume that a single neural model or an ensemble of neural models will produce an effective solution. In reality, it is difficult for any one model to learn all of the varied ways in which answers correspond to questions presented by the user. Due to the high cost of training and evaluating neural models, researchers often don't consider more sophisticated combinations of models, or ensembles with non-neural components. This movement away from the multi-strategy, multi-component approach that reached its zenith in IBM Watson is unfortunate, because it has focused the QA field on just a few, artificially created datasets that are comparatively easy for neural QA approaches.

It is ironic that the best-performing automatic QA system in the LiveQA evaluations (Wang and Nyberg 2015b, 2016, 2017) combined sophisticated neural models with an optimized version of the classic BM25 algorithm; neither the neural model nor BM25 was competitive by itself, but the combination of these two algorithms provided the most effective solution for the Yahoo! Answers data set. While it is true that curating datasets which can be solved by neural methods has stimulated the development of more capable, sophisticated neural models, neural approaches still rely on hundreds of thousands of labeled examples, and do not perform well when (a) there is limited training data, (b) there is a large variance in the lengths of the question versus answer texts, and (c) there is little lexical overlap between question and answer texts (Wadhwa et al. 2018b, c).

8.5 Conclusion

As we have discussed in this chapter, the development of common interchange formats for language processing modules in the JAVELIN project (Lin et al. 2005; Mitamura et al. 2007; Shima et al. 2006) led to the use of common schemas in the NTCIR IR4QA embedded task (Mitamura et al. 2008), which we believe is the first example of a common QA evaluation using a shared data schema and automatic combination runs. Although it is expensive to use human evaluators to judge all possible combinations of systems, automatic metrics (such as ROUGE) can be used to find novel combinations that seem to perform well or better than the state of the art; this subset of novel systems can then be evaluated by humans. In the OAQA project (which followed JAVELIN at CMU), development participants began to create gold-standard datasets that include expected outputs for all stages in the QA pipeline, not just the final answer (Garduño et al. 2013). This allowed precise automatic evaluation and more effective error analysis, leading to the development of high-performance QA incorporating hundreds of different strategies in real time (IBM Watson) (Ferrucci et al. 2010). The OAQA approach was also used to evaluate and optimize several multi-strategy QA systems, some of which achieved state-of-the-art performance on the TREC Genomics datasets (2006 and 2007) (Yang et al. 2013) and BioASQ tasks (2015–2018) (Chandu et al. 2017; Yang et al. 2015, 2016).

Although academic datasets in the QA field have recently focused on specific parts of the QA task (such as answer sentence and answer span selection) (Rajpurkar et al. 2016, 2018) which can be solved by a single deep learning or neural architecture, systems which achieve state-of-the-art performance on messy, real-world datasets (such as Jeopardy! or Yahoo! Answers) must employ a multi-strategy approach. For example, neural QA components were combined with classic information-theoretic algorithms (e.g., BM25) to achieve the best automatic QA system performance on the TREC LiveQA task (2015–2017) (Wang and Nyberg 2015a, b, 2016, 2017), which was based on a Yahoo! Answers community QA dataset. It is our expectation that a path to more general QA performance will be found by upholding the tradition of multi-strategy, multi-component evaluations pioneered by NTCIR. In our most recent work, we have tried to extend the state of the art in neural QA by performing comparative evaluations of different neural QA architectures across QA datasets (Wadhwa et al. 2018a), and we expect that future work will also focus on how to curate the most challenging (and realistic) datasets for real-world QA tasks (Wadhwa et al. 2018c).

Acknowledgements We would like to thank to the editors and to the past organizers and participants of the NTCIR ACLIA QA tasks. Special thanks go to Hideki Shima, who worked on CMU's Javelin QA system to develop the component-based evaluation and helped to organize the ACLIA tasks. We also thank the other students and staff who contributed to the JAVELIN, ACLIA, OAQA, and LiveQA projects.

References

Andrews C (2011) Ibm announces eight universities contributing to the watson computing system's development. PR Newswire. https://tinyurl.com/yxsmx8q5

Asahara M, Matsumoto Y (2003) Japanese named entity extraction with redundant morphological analysis. In: Proceedings of the 2003 human language technology conference of the North American chapter of the association for computational linguistics. pp 8–15. https://www.aclweb.org/anthology/N03-1002

Chandu KR, Naik A, Chandrasekar A, Yang Z, Gupta N, Nyberg E (2017) Tackling biomedical text summarization: OAQA at bioasq 5b. In: Cohen KB, Demner-Fushman D, Ananiadou S, Tsujii J (eds) BioNLP 2017, Association for Computational Linguistics, Vancouver, Canada. pp 58–66. 10.18653/v1/W17-2307. https://doi.org/10.18653/v1/W17-2307

Ferrucci D, Lally A, Verspoor K, Nyberg E (2009a) Unstructured information management architecture (uima) version 1.0. OASIS Standard. OASIS

Ferrucci D, Nyberg E, Allan J, Barker K, Brown E, Chu-Carroll J, Ciccolo A, Duboue P, Fan J, Gondek D et al (2009b) Towards the open advancement of question answering systems. IBM, IBM Res Rep, Armonk, NY

Ferrucci D, Brown E, Chu-Carroll J, Fan J, Gondek D, Kalyanpur AA, Lally A, Murdock JW, Nyberg E, Prager J et al (2010) Building watson: an overview of the deepqa project. AI Magazine 31(3):59–79

Garduño E, Yang Z, Maiberg A, McCormack C, Fang Y, Nyberg E (2013) CSE framework: a uima-based distributed system for configuration space exploration. In: Klügl P, Eckart de Castilho R, Tomanek K (eds) Proceedings of the 3rd workshop on unstructured information management architecture, vol 1038. CEUR-WS.org, CEUR Workshop, Darmstadt, Germany, pp 14–17. http://ceur-ws.org/Vol-1038/paper_10.pdf

Lin F, Shima H, Wang M, Mitamura T (2005) CMU JAVELIN system for NTCIR5 CLQA1. In: Kando N (ed) Proceedings of the Fifth NTCIR workshop meeting on evaluation of information access technologies: information retrieval, question answering and cross-lingual information access, NTCIR-5, National Center of Sciences, National Institute of Informatics (NII), Tokyo, Japan. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings5/data/CLQA/NTCIR5-CLQA1-LinF.pdf

Lin J, Demner-Fushman D (2006) Will pyramids built of nuggets topple over? In: Proceedings of the human language technology conference of the NAACL, main conference. Association for Computational Linguistics, New York, USA, pp 383–390. https://www.aclweb.org/anthology/N06-1049

Mitamura T, Lin F, Shima H, Wang M, Ko J, Betteridge J, Bilotti MW, Schlaikjer AH, Nyberg E (2007) JAVELIN III: cross-lingual question answering from japanese and chinese documents. In: Kando N (ed) Proceedings of the 6th NTCIR workshop meeting on evaluation of information access technologies: information retrieval, question answering and cross-lingual information access, NTCIR-6. National Center of Sciences, Tokyo, Japan, National Institute of Informatics (NII). http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings6/NTCIR/51.pdf

Mitamura T, Nyberg E, Shima H, Kato T, Mori T, Lin C, Song R, Lin C, Sakai T, Ji D, Kando N (2008) Overview of the NTCIR-7 ACLIA tasks: advanced cross-lingual information access. In: Kando N (ed) Proceedings of the 7th NTCIR workshop meeting on evaluation of information access technologies: information retrieval, question answering and cross-lingual information access, NTCIR-7. National Center of Sciences, National Institute of Informatics (NII), Tokyo, Japan. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings7/pdf/NTCIR7/C1/CCLQA/01-NTCIR7-OV-CCLQA-MitamuraT.pdf

Rajpurkar P, Zhang J, Lopyrev K, Liang P (2016) Squad: 100,000+ questions for machine comprehension of text. arXiv:160605250

Rajpurkar P, Jia R, Liang P (2018) Know what you don't know: Unanswerable questions for squad. arXiv:180603822

Shima H, Wang M, Lin F, Mitamura T (2006) Modular approach to error analysis and evaluation for multilingual question answering. In: Calzolari N, Choukri K, Gangemi A, Maegaard B, Mariani J, Odijk J, Tapias D (eds) Proceedings of the fifth international conference on language resources and evaluation, LREC 2006. European Language Resources Association (ELRA), Genoa, Italy, pp 1143–1146. http://www.lrec-conf.org/proceedings/lrec2006/pdf/782_pdf.pdf

Voorhees EM (2003) Overview of the TREC 2003 question answering track. In: Voorhees EM, Buckland LP (eds) Proceedings of The twelfth text Retrieval conference, TREC 2003, vol Special Publication 500-255. National Institute of Standards and Technology (NIST), Gaithersburg, Maryland, USA, pp 54–68. http://trec.nist.gov/pubs/trec12/papers/QA.OVERVIEW.pdf

Voorhees EM (2004) Overview of the TREC 2004 question answering track. In: Voorhees EM, Buckland LP (eds) Proceedings of the thirteenth text retrieval conference, TREC 2004, vol Special Publication 500-261. National Institute of Standards and Technology (NIST), Gaithersburg, Maryland, USA. http://trec.nist.gov/pubs/trec13/papers/QA.OVERVIEW.pdf

Wadhwa S, Chandu KR, Nyberg E (2018a) Comparative analysis of neural QA models on squad. http://arxiv.org/abs/1806.06972

Wadhwa S, Chandu KR, Nyberg E (2018b) Comparative analysis of neural QA models on squad. In: Choi E, Seo M, Chen D, Jia R, Berant J (eds) Proceedings of the workshop on machine reading for question answering@ACL 2018. Association for Computational Linguistics, Melbourne, Australia, pp 89–97. 10.18653/v1/W18-2610. https://www.aclweb.org/anthology/W18-2610/

Wadhwa S, Embar V, Grabmair M, Nyberg E (2018c) Towards inference-oriented reading comprehension: parallelQA. http://arxiv.org/abs/1805.03830

Wang D, Nyberg E (2015a) CMU OAQA at TREC 2015 liveqa: discovering the right answer with clues. In: Voorhees EM, Ellis A (eds) Proceedings of the twenty-fourth text retrieval conference, TREC 2015, vol Special Publication 500-319. National Institute of Standards and Technology (NIST), Gaithersburg, Maryland, USA. http://trec.nist.gov/pubs/trec24/papers/oaqa-QA.pdf

Wang D, Nyberg E (2015b) A long short-term memory model for answer sentence selection in question answering. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing of the Asian federation of natural language processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers. The Association for Computer Linguistics, pp 707–712. http://aclweb.org/anthology/P/P15/P15-2116.pdf

Wang D, Nyberg E (2016) CMU OAQA at TREC 2016 liveqa: An attentional neural encoder-decoder approach for answer ranking. In: Voorhees EM, Ellis A (eds) Proceedings of the twenty-fifth text retrieval conference, TREC 2016, Gaithersburg, Maryland, USA, November 15–18, 2016, National Institute of Standards and Technology (NIST), vol Special Publication 500-321. http://trec.nist.gov/pubs/trec25/papers/CMU-OAQA-QA.pdf

Wang D, Nyberg E (2017) CMU OAQA at TREC 2017 liveqa: a neural dual entailment approach for question paraphrase identification. In: Voorhees EM, Ellis A (eds) Proceedings of the twenty-sixth text retrieval conference, TREC 2017, Gaithersburg, Maryland, USA, November 15–17, 2017, National Institute of Standards and Technology (NIST), vol Special Publication 500-324. https://trec.nist.gov/pubs/trec26/papers/CMU-OAQA-QA.pdf

Yang Z (2017) Analytics meta learning. PhD thesis, Carnegie Mellon University, 5000 Forbes Avenue

Yang Z, Garduño E, Fang Y, Maiberg A, McCormack C, Nyberg E (2013) Building optimal information systems automatically: configuration space exploration for biomedical information systems. In: He Q, Iyengar A, Nejdl W, Pei J, Rastogi R (eds) 22nd ACM international conference on information and knowledge management, CIKM'13, San Francisco, CA, USA, October 27 November 1, 2013, ACM, pp 1421–1430. https://doi.org/10.1145/2505515.2505692

Yang Z, Gupta N, Sun X, Xu D, Zhang C, Nyberg E (2015) Learning to answer biomedical factoid & list questions: OAQA at bioasq 3b. In: Cappellato L, Ferro N, Jones GJF, SanJuan E (eds) Working notes of CLEF 2015—conference and labs of the evaluation forum, Toulouse, France, September 8–11, 2015., CEUR-WS.org, CEUR Workshop Proceedings, vol 1391. http://ceur-ws.org/Vol-1391/114-CR.pdf

Yang Z, Zhou Y, Nyberg E (2016) Learning to answer biomedical questions: OAQA at BioASQ 4B. In: Proceedings of the fourth BioASQ workshop, association for computational linguistics, Berlin, Germany, pp 23–37. https://doi.org/10.18653/v1/W16-3104, https://www.aclweb.org/anthology/W16-3104