JAVELIN I and II Systems at TREC 2005

Eric Nyberg, Robert Frederking, Teruko Mitamura, Matthew Bilotti, Kerry Hannan,
Laurie Hiyakumoto, Jeongwoo Ko, Frank Lin, Lucian Lita, Vasco Pedro, Andrew Schlaikjer

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3891

{ehn,ref,teruko,mbilotti,khannan, hyaku,jko,frank,llita,vasco,hazen}+@cs.cmu.edu

1 Introduction

The JAVELIN team at Carnegie Mellon University submitted three question-answering runs for the TREC 2005 evaluation. The JAVELIN I system was used to generate a single submission to the main track, and the JAVELIN II system was used to generate two submissions to the relationship track. In the sections that follow, we separately describe each system and the submission(s) it produced, and conclude with a brief summary.

2 JAVELIN I: Main Track Run

The JAVELIN I system integrates a set of modules that perform various question-answering tasks, such as question analysis, document and passage retrieval, answer candidate extraction, answer selection, answer merging, and planning [16]. For the TREC2005 main QA task, our goal was to incorporate a new Answer Merger module, an extended Expert Information Extractor (IX) incorporating Answer Projection, and a reimplemented Java version of the proximity-based extractor (Light IX) originally developed for the multilingual version of JAVELIN [12]. In addition, several of the existing JAVELIN modules (Question Analyzer, Retrieval Strategist, and Planner) had undergone significant re-engineering since we last participated in TREC, and TREC 2005 provided an opportunity to test them on unseen data. We also retrained the support-vector machine (SVM) and finite-state transducer (FST) extractors in an effort to improve their performance.

2.1 Components used in the TREC evaluation

1. **Question Analyzer (QA)**. The Question Analyzer's primary functions include answer type classification and keyword selection. Answer types are now classified using the output of the RASP parser [3]. Using syntactic constituent boundaries from RASP, the QA can determine which constituents are important in selecting the answer type using a set of hand-coded rules. The desired answer type often corresponds to a noun phrase in the parse tree for the question. A mapping from tokens to answer types (using WordNet [6] as a resource) determines the hypothesized answer type for the entire question. The mapping is generated from WordNet hypernym relationships assisted by hand-coded rules.

   Keyword selection depends heavily on the recognition of named entities and significant phrases that can be used to find relevant documents. Not only are possible keywords verified in the WordNet lexicon; part of speech and lemma information assist in the choice of an appropriate grammatical form, and pruning of irrelevant keywords.

2. **Retrieval Strategist (RS) and Retrieval Executor (RE)**. This year, we have made a significant refinement of JAVELIN's retrieval architecture, factoring our existing Retrieval Strategist into a two-layer retrieval architecture. In the new architecture, the formulation of queries has been decoupled from the actual retrieval process. This separation supports number of retrieval modules, one per collection, fed by a single query formulation module, to enable federated search across distributed resources.

In place of the original Retrieval Strategist, there are now two modules. The first of these modules is a complete replacement for the query formulation functionality in the original Retrieval Strategist, and continues to bear that name. The other module is the Retrieval Executor, a thin wrapper around the Lemur IR engine[1] that is responsible for executing queries and retrieving documents.

The query formulation algorithm implemented by the Retrieval Strategist is one of gradual relaxation. Given a Question Analyzer output containing a set of keywords and an expected answer type, the RS formulates an ordered sequence of queries, where the first query is the narrowest or most specific query, and each successive query is broader / less specific than the previous one. At each step in the sequence, the query is relaxed along one or more dimensions, which include the size of the window that contains the keywords and the keyword ordering constraint. Windowing constraints for keywords identified as phrases and proper names are also relaxed, according to a different schedule. The later queries in the sequence are also relaxed by dropping keywords and/or the expected answer type from the query.

The primary purpose of the Retrieval Executor is to execute a sequence of queries, starting from the narrowest one. The RE concatenates the ranked document lists returned for each query until it has collected the requested number of documents, or until it has exhausted the query sequence. The RE also supports a socket interface for direct querying of the document collection, and responds directly to requests for document or passage text, or for corpus statistics.

3. **Answer Extractors**. The system includes a variety of Information Extractor (IX) modules, which implement different extraction algorithms and vary in their utility across different answer types. The simplest extractor is a proximity-based extractor (Light IX) whose task is to compute a non-linear distance function between the keywords and a candidate answer. We have enhanced our support vector machine-based extractor (SVM IX) by re-training it using additional semantic and structural features and using larger datasets of training questions. The SVM IX tries to discriminate between correct answers and incorrect ones based on local semantic and syntactic context. A finite state transducer-based extractor (FST IX) was used to incorporate extraction patterns - part of which were created manually, part of which were generalized automatically - and learn their precision with respect to each answer type. We also integrated a Java-based version of the proximity-based extractor (LIGHT2 IX) originally developed as part of the multilingual JAVELIN system.

   In addition to statistical extractors, we have incorporated an additional extractor (Expert IX) that combines various available resources [10] into a single high precision, low recall extractor. Currently, the Expert IX is able to find answers in resources such as various gazetteers and WordNet. From gazetteers, we extract very specific information such as structured information about planets, states and countries (e.g. state flag, diameter of Jupiter). The Expert IX also accesses WordNet [14] to perform limited reasoning based on local semantic information (hypernymy, synonymy, etc.) viewed as a simple semantic graph. The Expert IX handles factoid questions with high precision and can also handle definitional questions by extracting exact definitions and profiles from its resources. Similarly to the BBN system [21], answers/profiles are then used to query the local corpus and find similar snippets of text. In future work, we plan to enhance the coverage of the Expert IX while maintaining its high precision.

4. **Answer Generator (AG)**. The Answer Generator is responsible for producing a ranked list of answers from the set of answer candidates produced by the IX modules. The AG incorporates three steps: answer normalization, answer clustering and answer filtering. Answer normalization canonicalizes the answer candidates into type-specific formats to find redundant or complementary answers. The normalized answers are grouped into clusters, given the assumption that each candidate in the cluster is independent and equally weighted. Answer validation uses gazetteers to filter out invalid answers. When no adequate answer can be found after filtering, the AG the Planner, which may select a different strategy (e.g., application of a different IX module).

5. **Answer Merger (AM)**. The Answer Merger combines the answers from multiple extraction strategies. Answer merging for question answering is comparable to the merging of ranked lists from multiple search engines into a single list, e.g. via the Metasearch algorithms [1]. We incorporated Metasearch for answer merging in JAVELIN through use of merging strategies such as combSum, combMNZ, linear combination and logistic regression.

   CombSum sums the scores of the input answers and CombMNZ (Multiply-by-number-Non-Zero) multiplies the sum of the scores with the number of non-zero answers. These two methods use the scores from the input

[1] See: http://www.lemurproject.org/

without rescaling, and do not require any training. Linear combination is a sum of the weighted scores. To decide the weights, we used each input system's performance as a weight. Logistic regression (a.k.a. maximum entropy) is a statistical regression technique to predict the probability of binary variables. It has been used to combine the documents from multiple search engines [19]. In question answering, maximum entropy has been used to combine multiple answer selection modules [4]. As the performance of answer selection modules depends on the answer type of the questions, we trained different logistic regression models for different answer types. As a logistic regression model performs better than the other approaches, logistic regression was used as the answer merging strategy in the TREC evalution.

6. **Answer Projection**. The system used for the TREC evaluation incorporated *answer projection*, which is the task of retrieving a document from the collection that supports a given answer [15]. This process was used to find supporting documents for answers supplied by the Expert IX, which makes use of a variety of ontologies and gazetteers to provide high-precision answers for a subset of the questions. The new Retrieval Strategist treats an answer projection request as a special case of query formulation, one in which the answer is included in every query in the sequence. The relaxation schedule for the windowing constraint on the keywords (including the answer term(s)) is accelerated, broading quickly to the size of the entire document.

7. **Planner**. As with our previous TREC system, control of the question-answering process is provided by the JAVELIN Planner. The planning process begins after an initial analysis of the question, which is translated into a planning problem describing the initial information state (features of the current question, including its classification as either a FACTOID, LIST, or OTHER question), and an information goal defined in terms of the expected answer type.

   For TREC 2005, the planning domain model was extended to include the new information extractors, an operator (action) for Answer Projection, and an Answer Merger operator which combines the results from three extractors. Planning parameter estimates for operator success likelihoods were also revised to reflect the current performance of the JAVELIN QA components on a validation subset of TREC 8-12 questions. [2] A list of all operators in the TREC 2005 planning domain is presented in Table 3.

   Because our main focus this year was to evaluate the new Answer Merger, the Planner's SELECT_ANSWER operator was implemented to give preference to any answers produced by the AM over the answers from a single IX. The results of a single IX are used only if the AM fails (i.e., fewer than three extractors produced candidates).

   In addition to selecting the source of the final answer, the Planner was also responsible for producing the different answer formats required by the three question categories (FACTOID, LIST, OTHER) comprising the QA task. This was accomplished with a set of very simple heuristics based on the confidence scores of the ranked answers. For all FACTOID questions, the top answer was returned. In the case of LIST and OTHER questions, the Planner displayed all answers greater than or equal to a confidence threshold $c = (0.5 * \textit{top answer confidence})$, unless this threshold resulted in just a single answer being returned. In such cases, the threshold was lowered to $c = (0.5 * \textit{rank 2 answer confidence})$.

2.2 TREC Main QA Track Results

A single TREC QA run was submitted for the QA track main task and the document set retrieved from the main task was submitted for the document ranking task. In the main QA task, the JAVELIN system achieved an average F score of 0.169 for the factoid questions. JAVELIN obtained 0.059 on the list questions and 0.015 on other questions. In the document ranking task, the average precision for all relevant documents was 0.1584. However, the system achieves its highest precision with a cutoff of 15 documents (23.87% precision and almost 28% recall).

2.3 Analysis

In the interim between performing the TREC QA evaluation and receiving our official scores, we conducted an internal performance analysis for a subset of the TREC 2005 question set. Since we focused on improving extraction quality for temporal and numeric questions, we did a detailed analysis of those answer types. Project members manually identified correct answers for temporal and numeric questions, along with at least one document containing

[2] We implemented additional operators to merge the results from varying numbers of etractors, but due to time constraints, these operators were not enabled during the TREC evaluation.

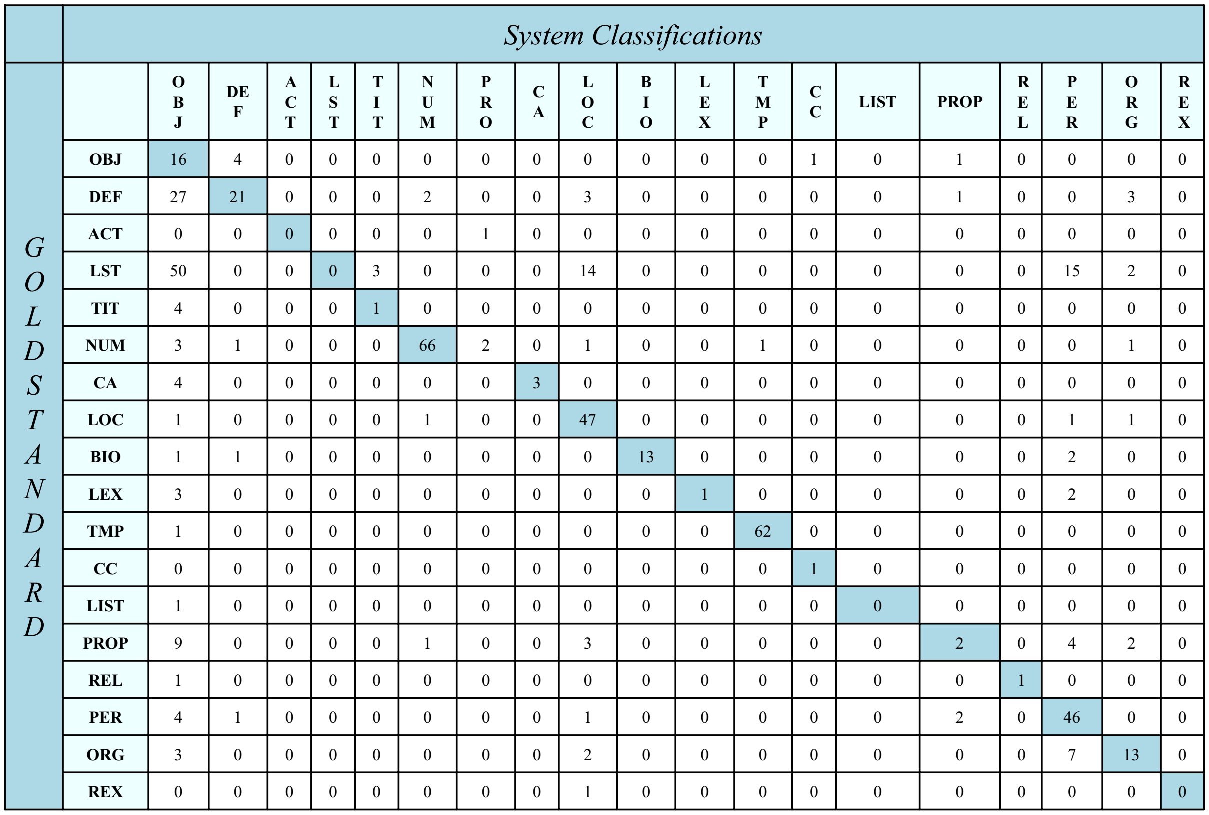
the answer. We then compared our manually generated answer key with the system's output to determine whether the system returned the correct answer. If not, we identified the point of failure. Table 4 summarizes the results of this analysis. For each failure point, we computed the number of questions that resulted in errorneous output. The results show that roughly equal percentages of the failure occurred during document retrieval, answer candidate extraction and answer selection.

The performance of JAVELIN was compared with the previous JAVELIN TREC run (TREC 2003). As can be seen in Table 5, this year's system improved performance on factoid questions by 30.8%.

To evaluate the performance of the Question Analyzer, we generated an answer type classification confusion matrix. Figure 1 shows the answer type classification confusion matrix for the TREC 2005 evaluation. The accuracy of answer type classification for the TREC run we submitted was 0.553. On the training data (TREC8-12), the accuracy of answer type classification was 0.763.

We also measured the performance of each component for the training questions in TREC8-12. Table 6 shows the performance of five extractors. The outputs from the Question Analyzer and Retrieval Strategist were reused to test multiple IXes. Macro-average assigns an equal weight to each category, regardless of how rare or common a category is. So macro average considers all categories as "equal". Micro-average assigns an equal weight to each question/document (more generic category instance), favoring performance on larger categories [22]. "Any" provides the percentage of questions with at least one good (matching key) document (in RS) and answer (in IX) in the set.

These results were used to train the Answer Merger. To decide a merging strategy for the TREC 2005 evaluation, the combination of four extraction strategies (FST, LIGHT, SVM and Expert IX) was tested with different merging techniques. The results show that the logistic regression model outperformed other merging methods and improved the system performance by 13.3% over the best stand-alone answer strategy and by 7.8% over a linear combination model. On the other hand, combSum and combMNZ did not improve the performance at all.



		System Classifications																		
		OBJ	DEF	ACT	LST	TIT	NUM	PRO	CA	LOC	BIO	LEX	TMP	CC	LIST	PROP	REL	PER	ORG	REX
GOLD STANDARD	OBJ	16	4	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
	DEF	27	21	0	0	0	2	0	0	3	0	0	0	0	0	1	0	0	3	0
	ACT	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	LST	50	0	0	0	3	0	0	0	14	0	0	0	0	0	0	0	15	2	0
	TIT	4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	NUM	3	1	0	0	0	66	2	0	1	0	0	1	0	0	0	0	0	1	0
	CA	4	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
	LOC	1	0	0	0	0	1	0	0	47	0	0	0	0	0	0	0	1	1	0
	BIO	1	1	0	0	0	0	0	0	0	13	0	0	0	0	0	0	2	0	0
	LEX	3	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	0	0
	TMP	1	0	0	0	0	0	0	0	0	0	0	62	0	0	0	0	0	0	0
	CC	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	LIST	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	PROP	9	0	0	0	0	1	0	0	3	0	0	0	0	0	2	0	4	2	0
	REL	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	PER	4	1	0	0	0	0	0	0	1	0	0	0	0	0	2	0	46	0	0
	ORG	3	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	7	13	0
	REX	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Figure 1: Answer type classification confusion matrix (TREC 2005)

2.4 Post-TREC Extensions

1. **Answer Merger (AM)**. After the TREC evaluation, we did further investigation on the performance of the Answer Merger when using a logistic regression model. As we have five answer extractors, we generated 26 combinations of the extractor outputs and tested the AM with each combination. The results show that the AM produced the best score when combining only four extractors (Expert, FST, Light and Light2 IX). It improved the answer merging performance by 26% over the best stand-alone result (Light IX). However when combining all of the extractors, the performance improved only by 21.1%. Even though it performs relatively better than the other extractors, the SVM IX did not improve the performance of answer merging. This is mostly because the SVM IX does not produce very reliable confidence scores for certain answer types.

   To decide whether we should just exclude the SVM in answer merging, we compared the performance of the AM for each answer type. This comparison shows that adding the SVM IX improved the performance of numeric-expression and temporal questions, but did not improve the performance of location, person-name and object questions. This information can be useful to a QA system when making strategy decisions, such as selecting the combination of answer lists to maximize performance for the current answer type. For example, when processing numeric-expression and date questions, the Planner calls all five extractors. On the other hand, when processing location and person-name questions, the Planner calls only four extrators. This indicates that multi-strategy answer merging is important in multi-strategy QA systems. For more detailed analysis, see [9].

   We plan to do more experiments by incorporating regularization. Si and Callan (2005) have recently shown that regularized logistic regression improves merging for multilingual document lists, and we intend to investigate its application to answer merging in QA. In addition, we will extend the AM to support the multilingual JAVELIN QA system.

2. **Answer Generator**. As some of the extractors return too many answer candidates, answer selection has been a challenge, and it is generally difficult to identify the correct answer amongst many incorrect ones. To improve answer selection accuracy, we have combined evidence provided from three semantic resources: excerpts from Web documents, WordNet, and several gazetteers including the CIA World Factbook.

   For gazetteers we assigned the following confidence score for each answer candidate: 1.0 if gazetteers can identify the answer, 0.5 if the answer occurs in the gazetteer within the subcategory of the expected answer type (e.g., if the candidate ”Shanghai” is a city, given the question ”Which city in China has the largest number of foreign financial companies?”), 0.0 otherwise. The same approach was used for WordNet. A Web score is computed based on the heuristic approach presented by (Magnini et al., 2002) to analyze the text snippets returned by Google.

   As a preliminary experiment, we tested location and proper-name questions in TREC8-12 with combSum, linear combination, and logistic regression. The results show that linear combination was the best for the location questions. This improved the performance by 15.93%. For proper-name questions, logistic regression produced the best score and improved answer selection performance by 75.93% [9]. We are adding more resources such as Wikipedia and the EIJIRO dictionary to improved our answer selection performance and to support the multilingual JAVELIN system.

2.5 Future work

Time constraints limited our ability to fully exploit the new functionality provided by the new Retrieval Strategist and Answer Merger for TREC. In particular, the RS2 supports a filtering option that can be used to perform successive document retrieval, and the Answer Merger supports N-way merges, not just the 3-way merge used by the planner. Our expanded planning domain includes both a filtering operator and additional operators for N-way merges, and we continue to work on better tuning of planning parameters for all operators. We also continue to test variations of the AM that consider answer type when selecting the weights and merge strategies.

3 JAVELIN II: Relationship Track Runs

The JAVELIN II system design is composed of a collection of server modules which perform various high-level tasks in a typical question answering system pipeline. A Question Analysis module is provided input question text along with any contextual material and produces an analysis of the user's information need. This analysis is then used by an Information Retrieval engine, the Retrieval Strategist and Retrieval Executor, to produce a short list of ranked documents from a corpus of trusted material. An Information Extraction module takes the ranked list of documents, along with the initial question analysis output, and extracts from the document set more specific information related to the user's information need. Finally, an Answer Generation module filters the prioritized output from the Information Extraction module and synthesizes a final response to the user's query. More detail on system architecture can be found in [17].

The system architecture exposes precise module interface specifications, allowing various implementation strategies to be adopted on a per-module basis. This flexibility in module implementation also affords straight-forward adaptation of external software packages for use in the system. Of central concern to the JAVELIN II system is a uniform mechanism for the integration of data from multiple sources, which is achieved via the Annotations Database and Text Annotator frameworks[3].

3.1 JAVELIN II Components in TREC 2005

3.1.1 Question Analyzer

JAVELIN II's Question Analyzer module has been substantially modified from its initial design. In earlier versions of this module, the goal of question analysis was the identification of question and answer types of various specificity, along with a set of relevant keywords for an information retrieval stage of question processing. In addition to this process, the JAVELIN II question analysis module identifies target semantic predicates expected to be found in relevant answer texts. These target semantic predicates are enriched with entity type information and weighted alternate predicate hypotheses, generating a diverse set of relational matching criteria. These target semantic structures are then employed in later information filtering and ranking steps to arrive at answer candidates.

To produce a set of target semantic structures, a collection of specialized tools first create various layers of annotation on input question text. These layers of annotation are then merged to form a set of semantic structures, weighted based on the specificity of compositional annotation present in a given structure. Among the tools used to create annotations are Colorado University's ASSERT semantic parser [18], which users a statistical model derived from Propbank [11] data; the BBN IdentiFinder named-entity recognizer [2]; and ontological resources such as Princeton University's WordNet lexical database [6] and the CNS Ontology [7]. All annotations generated from these tools are collected in a single, unified Annotations Database, accessible from all other JAVELIN system modules, affording great flexibility to the weighting and prioritization of annotations during the creation of target semantic structures.

Overall semantic structure weights, along with more fine-grained weights applied to components of target semantic structures, influence information retrieval query formulation and backoff, as well as information extraction filtering and ranking procedures.

3.1.2 Retrieval Strategist and Retrieval Executor

These modules were reused from the JAVELIN I system described above. A more advanced set of information retrieval modules has been developed for future use in JAVELIN II, but because of the lack of full annotation coverage of the TREC evaluation corpus, we were unable to use these for our relationship track submissions. The more advanced modules use the semantic structures output by the Question Analysis module to form very specific queries against structural metadata indexed along with the original text of annotated corpora. Furthermore, the more advanced Retrieval Strategist module includes query backoff routines which action the priorities of the components of semantic structures.

3.1.3 Semantic Information Extractor (SemIX)

Once a document set has been selected by the Retrieval Strategist, candidate answer sentences are extracted and ranked from these documents by the Semantic Information Extractor (SemIX). This process mirrors that of question analysis, where target semantic structures are created from various layers of annotation on input text. In the SemIX, the text of documents reported by the Retrieval Strategist is annotated, and semantic structures are generated similarly to the process described in section 3.1.1. The annotation of document text is done online, during query processing, only if those documents have not already been annotated previously during offline corpus annotation

[3] Details on these frameworks is outside the scope of this paper, but information and resources are available at `http://durazno.lti.cs.cmu.edu/javelin_public/releases/`

runs. Significant computational resources are required to fully annotate the TREC corpus with our suite of tools, and not all documents were fully processed by the time the TREC Relationship QA track had begun. Because of the small number of Relationship track questions, selective online annnotation of corpus materials did not pose a significant problem. However, we were not able to apply more advanced information retrieval techniques in our TREC system because of the partially annotated state of the test corpus.

After completion of document set annotation, structural patterns are generated from the target semantic structures output by the Question Analyzer. These structural patterns are applied to all semantic structures created from the document set in a ranking procedure, where those semantic structures not meeting a minimal match threshold are removed from further consideration. For a single semantic structure in corpus materials, the ranking procedure combines evidence from all target semantic structures. Matching semantic structures are then collated with their containing sentences, and per-structure scores are combined to give sentence-level scores and a global relative ordering of candidate answer sentences. The ranked list of output sentences can be tens or hundreds of sentences long, depending on the amount of source material which matches some portion of the target semantic structures, and duplicate information is likely at this stage.

3.1.4 Answer Generator (AG)

The Answer Generator module is responsible for producing the set of nuggets that answer each question from the set of answer candidates produced by the SemIX. It does this by eliminating duplicates from the nuggets produced by the SemIX and returning the set of unique nuggets. The cut-off point was set at 15, and in all cases the cut off point was reached (15 nuggets were returned).

The identification of unique nuggets is done by taking each pair of nuggets and calculating the average between the Levenstein distance and Cosine Similarity. The Levenstein distance is a variation of the Edit distance and the purpose is to measure the difference between string at the character level, while the Cosine Similairity measure will focus on the string difference at the word level by using vector comparison. The current AG is an attempt to filter the nuggets produced by the SemIX and assure a maximum number of unique nuggets.

The other role of the AG is to format the results to conform to the TREC output specifications. The AG creates a file to which appends the nuggets in the correct form, using the Request Object to retrieve the question ID and the Document ID.

3.2 System Performance and Analysis

The JAVELIN II system was used to submit two runs for the relationship track. The first, fully-automatic run of the system failed to produce answers for six of the relationship questions, due to failure at the Question Analyzer stage. In JAVELIN II, it is essential that the question analysis process produce a semantic structure (key predicate(s) for the input; this is achieved using the ASSERT tool. In some cases, however ASSERT did not produce an output for a question. To address this shortcoming, a team member familiar with ASSERT's output manually labeled the semantic structures in all 25 of the test questions. The second, semi-automatic run submitted used this gold-standard question analysis as input to the rest of the JAVELIN II pipeline. The system was able to produce an answer for each of the questions in the second run.

Table 7 shows the performance of the two JAVELIN II runs with respect to that of the other track participants. The fully-automatic run, denoted by the run tag `CMUJAVSEM`, ranked eleventh out of eleven runs in terms of F(3) measure, and also in terms of precision and recall, each considered separately. The semi-automatic run, with the run tag `CMUJAVSEMMAN`, ranked ninth out of eleven runs in terms of both F(3) measure and precision. In terms of recall, the semi-automatic run placed tenth because the runs `RUN-9` and `RUN-3` tied for eighth place.

Given that `CMUJAVSEMMAN` was only 0.006 behind `RUN-9` and `RUN-3` in terms of recall, it is likely that the true recall of the three systems is actually quite similar. The difference lies in precision. `RUN-9`, with `RUN-3` not too far behind, was the leader in terms of precision. It was these two runs that returned relatively concise lists of nuggets for the questions, and the two of them outscored by a broad margin all of the other runs. Because these two systems were able to detect and refrain from returning non-relevant nuggets, even though they had performance similar to `CMUJAVSEMMAN` in terms of recall, they ranked higher than it in terms of F(3) measure.

`RUN-7` is an interesting outlier that ranked 10th in terms of both F(3) measure and recall, right between `CMUJAVSEMMAN` and `CMUJAVSEM`. Unlike `RUN-9` and `RUN-3`, `RUN-7` was characterized by returning enormous nugget lists for certain questions. One extreme example from the official assessment of `RUN-7` is that credit was given on question 17 for matching a nugget on the answer key with their 11,119-th ranked nugget. `RUN-7` had excellent recall, scoring second, but on several occasions, matching nuggets were so far down on the ranked list, that very few users would ever be patient enough to find them. The precision score of RUN-7 suffered as a result, yet, interestingly CMUJAVSEM had a lower precision score than RUN-7, even though JAVELIN II returns a maximum of fifteen nuggets per sentence.

In the official evaluation, assessors were asked to read each system's ranked list of nuggets, and to match them manually against the questions' answer keys. Assessors are asked to use their understanding of natural langage to make positive matches between system responses and answer key nuggets without penalizing for different word usage or syntactic structure, or for paraphrase and rephrasing of the answer key nugget [20].

We recognize that this may be a difficult or tedious task for humans to perform, so we repeated it with an eye toward checking if any mistakes were made unifying the nuggets our system returned with those on the answer key. Although there were cases where we failed to understand how the assessors decided to draw the distinction between vital and okay nuggets, we did not attempt to second-guess these decisions when we repeated the evaluation. See [8] for some discussion of how the vital/okay distinction can affect evaluation.

While performing our evaluation, we identified 7 vital and 8 okay nuggets in our CMUJAVSEM run that the assessors did not give credit for. In CMUJAVSEMMAN, we found 3 vital, and 3 okay nuggets that the assessors did not give credit for. We developed a new set of judgments based on these changes, and, though not official, system performance evaluated against this judgment set gives an impression of an upper bound on our scores. With the new judgments, CMUJAVSEM scored 0.058 in precision, 01.62 in recall, and 0.131 in F(3) measure. CMUJAVSEMMAN scored 0.058 in precision, 0.163 in recall and 0.129 in F(3) measure. We noted that, under the new judgments, CMUJAVSEM would have ranked 7th and CMUJAVSEMMAN would have ranked 8th in terms of F(3) measure, followed by RUN-9, RUN-3 and RUN-7. CMUJAVSEM and CMUJAVSEMMAN would have tied for 9th in precision, with 11th place going to RUN-7. CMUJAVSEMMAN would have ranked 8th and CMUJAVSEM would have ranked 9th in recall, followed by RUN-9 and RUN-3 tied for 10th.

By analyzing the results of the CMUJAVSEMMAN run, we can draw some general conclusions about JAVELIN II's performance. There were several questions where our nuggets were on-topic, but simply not specific enough to match the nuggets in the answer key. Examples include: Q5, where our system returned nuggets relevant to the Chechnya issue, but didn't identify groups that supported the rebels; Q10, where JAVELIN II got nuggets about Ecuador, but not about drug interdiction efforts; Q15, where the system discusses China and Taiwan in a military sense, but fails to address whether pressure from Beijing has affected the sale of armaments to Taipei; and Q17, where JAVELIN II talks about Israel and China with respect to Middle East peace, but fails to discuss arms trading.

One specific area where JAVELIN II fell short of other systems was in the interpretation of constraints present in the question. Specifically, the higher-scoring systems properly retrieved a list of countries when asked, but this functionality was never explicitly included in JAVELIN II. A prime example where this shortcoming hurt JAVELIN II's performance was Q22, where our system should have returned a list of countries. Q23 is an even better example; not only did the system fail to extract country names, but it also failed to restrict the list of countries to those in South America, as the question asked. In response to a request for countries seeking nuclear capability, JAVELIN II named Syria, Libya, Iran and North Korea, which would have been reasonable answers, except that the system failed to note the constraint that only South American countries should be considered.

A general observation about performance across the entire track was that there were some questions that were just generally difficult to answer. For questions 2 and 20, only one of the systems got an answer, and for questions 2, 3, 7, 17, 20, 23 and 24, fewer than half of the systems were able to come up with an answer. That said, the two JAVELIN II runs featured the most questions for which no answer key nuggets were matched. Followed by RUN-3 and RUN-9, which each missed more than 10 questions, all other runs were able to get credit for more than fifteen questions.

3.3 Evaluating Relationship QA as a Ranked List of Nuggets

Prior to the release of the official Relationship QA track results, we performed an in-house evaluation. Given that JAVELIN II focused on ranking of relevant nuggets retrieved from the corpus, it seemed natural to evaluate our ranked nugget lists directly. To perform any sort of evaluation on our nuggets, we were going to need relevance judgments.

Three team members were asked to read the ranked lists from both system runs and to assign binary relevance judgments to each nugget based on their own notion of information need and relevance. No guidelines were used, and no attempt was made to maximize inter-coder agreement. Instead, we wanted to obtain samples of how the nuggets compared with three different information needs. Using these binary judgments, we were able to assign a score to each nugget, in the range from zero to three, corresponding to the number of different information needs

Table 1: Distribution of Nuggets by Score

Run	Score 0	Score 1	Score 2	Score 3	Total
fully-automatic	193 (67.72%)	46 (16.14%)	33 (11.58%)	13 (4.56%)	285
semi-automatic	247 (65.87%)	50 (13.33%)	43 (11.47%)	35 (9.33%)	375
overall	440 (66.67%)	96 (14.55%)	76 (11.52%)	48 (7.27%)	660

Table 2: MRR Comparison of System Runs

Fully-Automatic			Semi-Automatic	
0.4457	0.5338	$S \geq 1$	0.7040	0.7509
0.3860	0.4553	$S \geq 2$	0.3760	0.4316
0.1538	0.2023	$S \geq 3$	0.2733	0.2965

that judged the nugget relevant.

Table 1 shows how this score can be used to analyze the distributions of nuggets in each of the two runs. The semi-automatic run, using the gold-standard question analysis, retrieved almost 5% more Score 3 nuggets and approximately the same number of Score 2 nuggets, when compared with the fully-automatic run.

In Table 2, we use a simple MRR metric that averages the reciprocal rank of the first nugget with a score of at least S across all of the questions. There were no nuggets retrieved for each of the six questions (8, 14, 19, 21, 24 and 25) that the automatic question analysis was not able to process, so in the second from leftmost column in the table, the average is taken over 19 questions. In the leftmost column, the average is taken over all 25 questions, where only score zero nuggets were assumed to have been retrieved for these six questions. The far right column in the table shows a score calculated for the semi-automatic run over only those 19 questions that the fully-automatic run was able to process. From top to bottom in the table, the minimum score of the nugget (S) increases, so that in the top row, the first nugget with $S \geq 1$ is chosen for the purposes of computing the score, and on the bottom row, the first nugget with $S \geq 3$ is chosen.

3.4 Future Work

Planned extensions to existing JAVELIN II modules address a number of shortcomings of the modules used for this year's relationship track, as well as extensions which address problems in parallel domains, such as scenario-based question answering.

Because question analysis requires high recall from semantic parsing of input question text to recover even partial predicate structures, efforts will be made to develop a more robust semantic parsing tool, capable of greater coverage and higher throughput than the semantic parsing tool currently used. Not only will higher coverage of question text be beneficial, but application of a more robust semantic parser to corpus materials is also expected to greatly increase system performance.

More ontological data will also be incorporated into JAVELIN II's semantic structures with the completion of an interface to the Scone ontology [5]. This resource will provide JAVELIN with greater hierarchical type information on entities found in text, as well as allow semantic structures generated from question text to be enriched with alternate and parallel semantic relations, conditioned on the question domain.

Anaphora resolution technology will be applied both to input question text as well as corpus materials in order to improve accuracy of semantic structure creation and matching.

Our information retrieval engine will move from Lemur to Indri [13], the next generation search engine from the Lemur project, which will allow more precise structured search of annotated corpus materials. This we expect to filter out many false positives from Retrieval Strategist results, allowing the Information Extractor to find and report higher-scoring semantic structures.

To refine the bounds on selected text returned by the Information Extractor from a whole sentence, to a specifc phrase or term, the semantic structure will evolve to include terms representing unbound variables. During information extraction, these unbound variables will be bound to specific elements of matching semantic structures, and be reported along with their containing sentences.

Improvements to semantic structure scoring metrics will also be investigated, utilizing link analysis techniques on corpus semantic structures. Through greater understanding of various corpus statistics for semantic structures, better weighting of the importance of relation and entity types, as well as specific instances of relations and entities present in corpus materials, may be possible.

4 Summary

There are a few general observations which can be drawn from the performance of the JAVELIN I and II systems in the TREC 2005 evaluations. First, it seems evident that the performance of the statistical-based extractors and the pattern-based question analysis used in JAVELIN I are weak points for that system. Both of these components show poorer performance on unseen data vs. training data, and the accuracy of document and answer retrieval is impacted accordingly. Even when specialized statistical extractors are developed for different answer types, it seems that much larger amounts of training data would be required to realize an upper bound on extractor performance. Although resource-based extractors (e.g. Expert IX) can provide high-precision answers, in practice there were few TREC questions that were answered by JAVELIN I topic-specific resources, which placed greater emphasis on statistical extractors. Inclusion of a large and diverse number of topic-specific resources is required to realizegains from resource-based extractors in TREC-style evaluations.

The first run of the JAVELIN II system, with its more semantics-based approach, indicates that more work is required for effective question analysis, in particular to identify key semantic structures which should be sought in the target corpus. Another failure point for the JAVELIN II system was the lack of semantics-based indexing and retrieval, which we have already taken steps to remedy since the TREC evaluation. Even if a corpus has been annotated with semantic information, if that information is not indexed for retrieval at run-time the system must resort to standard keyword-based retrieval followed by semantic analysis of what can be gleaned from the document set. We expect that our TREC 2005 relationship track results will provide a useful baseline for testing the effectiveness of semantic indexing and retrieval, as well as the use of additional general and domain-specific ontological information for answer extraction.

A Tables

Operator Name	Description
RETRIEVE_DOCUMENTS	Calls the RS. Applicable when there is an active question.
EXTRACT_CANDIDATES	Calls one of the four conventional information extractors (SVM, FST, LIGHT, LIGHT2). Applicable when there is an active question and a document set. Outcome likelihoods are generated dynamically, conditioned on the type of extractor and expected answer type.
CONSULT_KNOWLEDGE_BASE	Calls the EXPERT extractor. Applicable when there is an active question to answer. Outcome likelihoods are generated dynamically, conditioned on the expected answer type.
RANK_KB_CANDIDATES	Calls the AG to rank candidates produced by the EXPERT extractor. Applicable when the EXPERT extractor has produced candidates.
RANK_IX_CANDIDATES	Calls the AG to rank candidates produced by one of the four conventional extractors. Applicable when such candidates exist. Outcome likelihoods are conditioned on the expected answer type and extractor which generated the candidates.
PROJECT_ANSWER	Calls the RS to retrieve documents that support the EXPERT candidates. Applicable when the final answer selected comes from the EXPERT IX.
MERGE_ANSWERS3	Calls the AM to merge candidate sets from three extractors. Applicable whenever at least three extractors have produced candidates (SVM, FST, LIGHT, LIGHT2, EXPERT).
SELECT_ANSWER	Chooses an answer to display, giving preference to ranked answer lists produced by the AM over ranked answers from a single source. Applicable when at least one ranked answer list exists.
DISPLAY_ANSWER	Completes the planning session by returning the selected answer to the GUI. Applicable once an answer is selected.

Table 3: TREC 2005 planning domain operators (actions).

Failure point	No. of questions	Question Percentage
a. system failed to retrieve any documents containing the answer	26	23.0%
b. system failed to extract any correct answer candidates	30	26.5%
c. system failed to select the correct answer candidate[4]	29	25.7%
d. module crashed or experienced unrecoverable error	1	0.9%
e. no failure	27	23.9%

Table 4: Failure analysis of temporal and numeric questions

	TREC 2005						TREC 2003	
Answer type	Questions	Accuracy	R	W	X	U	Quesions	Accuracy
action	1	0.00	0	1	0	0		
causal-antecedent	8	0.00	0	8	0	0	29	0.00
causal-consequence	3	0.00	0	3	0	0		
definition	7	0.14	1	6	0	0	5	0.00
lexicon	6	0.00	0	6	0	0	21	0.14
list	1	0.00	0	1	0	0		
location	56	0.20	11	36	5	4	57	0.35
numeric-expression	81	0.14	11	66	2	2	109	0.15
object	18	0.17	3	15	0	0	43	0.00
organization-name	26	0.19	5	19	1	1	12	0.00
person-bio	1	0.00	0	1	0	0		
person-name	57	0.23	13	41	3	0	36	0.17
proper-name	22	0.18	4	18	0	0	37	0.03
regex	2	0.00	0	2	0	0		
relation	1	0.00	0	1	0	0	1	0.00
temporal	68	0.19	13	49	4	2	47	0.13
title	4	0.00	0	3	1	0	16	0.19
ALL	362	0.17	61	276	16	9	413	0.13

Table 5: Performance of JAVELIN 2005. JAVELIN scores are compared with JAVELIN 2003 according to the answer type.

	Final Scores		QA		RS		IX			AG	
IX type	MRR	TREC	Micro F1	Macro F1	Any	Avg Prec	Any	Top 5	MRR	TREC	MRR
Expert	0.036	0.033	0.868	0.622	79.91%	0.312	4.51%	4.51%	0.04	0.033	0.036
Light2	0.279	0.223	0.868	0.622	79.91%	0.312	59.01%	47.64%	0.328	0.223	0.279
FST	0.139	0.128	0.868	0.622	79.91%	0.312	21.03%	20.60%	0.169	0.128	0.139
SVM	0.261	0.215	0.868	0.622	79.91%	0.312	60.30%	41.42%	0.266	0.215	0.261
Light	0.297	0.245	0.868	0.622	79.91%	0.312	63.73%	50.43%	0.364	0.245	0.297

Table 6: Performance of JAVELIN components on 665 questions. The questions were randomly chosen among TREC8-12 corpus for the training

Rank	F(3) Measure	Precision	Recall
1st	RUN-2 (0.282)	RUN-9 (0.159)	RUN-2 (0.451)
2nd	RUN-1 (0.230)	RUN-3 (0.150)	RUN-7 (0.431)
3rd	RUN-10 (0.220)	RUN-2 (0.078)	RUN-4 (0.345)
4th	RUN-4 (0.216)	RUN-11 (0.075)	RUN-1 (0.344)
5th	RUN-11 (0.190)	RUN-4 (0.074)	RUN-10 (0.338)
6th	RUN-6 (0.163)	RUN-6 (0.073)	RUN-11 (0.284)
7th	RUN-9 (0.120)	RUN-1 (0.070)	RUN-6 (0.223)
8th	RUN-3 (0.119)	RUN-10 (0.065)	RUN-9/RUN-3 (0.129)
9th	CMUJAVSEMMAN (0.096)	CMUJAVSEMMAN (0.047)	
10th	RUN-7 (0.086)	RUN-7 (0.040)	CMUJAVSEMMAN (0.123)
11th	CMUJAVSEM (0.061)	CMUJAVSEM (0.032)	CMUJAVSEM (0.085)

Table 7: Relationship QA Track Results: JAVELIN II's fully-automatic run is designated `CMUJAVSEM`, and the semi-automatic run is designated `CMUJAVSEMMAN`

References

[1] Javed A. Aslam and Mark Montague. Models for metasearch. In *Proceedings of SIGIR*, 2001.

[2] D. M. Bikel, R. L. Schwartz, and R. M. Weischedel. An algorithm that learns what's in a name. *Machine Learning, vol. 34, no. 1-3*, 1999.

[3] E. Briscoe and J. Carroll. Robust accurate statistical annotation of general text. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, 2002.

[4] Abdessamad Echihabi, Ulf Hermjakob, Eduard Hovy, Daniel Marcu, Eric Melz, and Deepak Ravichandran. Multiple-engine question answering in textmap. In *TREC*, 2003.

[5] Scott E. Fahlman. *Scone User Manual*, 2005.

[6] Fellbaum. Wordnet - an electronic lexical database. 1998.

[7] The Center for Nonproliferation Studies. The cns ontology. 2004.

[8] Wesley Hildebrandt, Boris Katz, and Jimmy Lin. Answering definition questions with multiple knowledge sources. In *Proceedings of HLT/NAACL 2004*, 2004.

[9] Eric Nyberg Jeongwoo Ko, Laurie Hiyakumoto. Exploiting multiple semantic resources for answer selection. In *Proceedings of of LREC 2006*, 2006.

[10] Boris Katz, Jimmy Lin, Daniel Loreto, Wesley Hildebrandt, Matthew Bilotti, Sue Felshin, Aaron Fernandes, Gregory Marton, , and Federico Mora. Integrating web-based and corpus-based techniques for question answering. In *TREC*, 2003.

[11] Paul Kingsbury, Martha Palmer, and Mitch Marcus. Adding semantic annotation to the penn treebank. 2002.

[12] Frank Lin, Hideki Shima, Mengqiu Wang, and Teruko Mitamura. Cmu javelin system for ntcir5 clqa1. In *Proceedings of the 5th NTCIR Workshop*, 2005.

[13] D. Metzler and W. B. Croft. Combining the language model and inference network approaches to retrieval. *Information Processing and Management Special Issue on Bayesian Networks and Information Retrieval*, 40(5):735–750, 2004.

[14] G. A. Miller. Wordnet: A lexical database for english. *CACM, 38(11):39-41*, 1995.

[15] Gilad Mishne and Maarten de Rijke. Query formulation for answer projection. In *27th European Conference on Information Retrieval (ECIR'05)*, 2005.

[16] E. Nyberg, T. Mitamura, J. Callan, J. Carbonell, R. Frederking, K. Collins-Thompson, L. Hiyakumoto, Y. Huang, C. Huttenhower, S. Judy, J. Ko, A. Kupsc, L. V. Lita, V. Pedro, D. Svoboda, and B. Van Durme. The javelin question-answering system at trec 2003: A multi-strategy approach with dynamic planning. In *TREC12*, 2004.

[17] Eric Nyberg, Teruko Mitamura, Robert Frederking, Vasco Pedro, Matthew Bilotti, Andrew Schlaikjer, and Kerry Hannan. Extending the javelin qa system with domain semantics. In *Proceedings of the 20th National Conference on Artificial Intelligence*, Pittsburgh, PA, June 2005. AAAI.

[18] Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, and Daniel Jurafsky. Shallow semantic parsing using support vector machines. 2004.

[19] Jacques Savoy, Anne Le Calv, and Dana Vrajitoru. Report on the trec-5 experiment: Data fusion and collection fusion. In *TREC*, 1996.

[20] Ellen M. Voorhees. Overview of the trec 2003 question answering track. In *Proceedings of TREC 2003*, 2003.

[21] Jinxi Xu, Ralph M. Weischedel, and Ana Licuanan. Evaluation of an extraction-based approach to answering definitional questions. In *SIGIR*, 2004.

[22] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, 1999.