An Evaluation Framework Based on Gold Standard Models for Definition Question Answering

Samir Kanaan and Jordi Turmo
TALP Research Center - Software Department
Universitat Politècnica de Catalunya
{skanaan,turmo}@lsi.upc.edu

Abstract

This paper presents a weak supervised evaluation framework for definition question answering (DefQA) called Solon. It automatically evaluates a set of DefQA systems using existing human definitions as gold standard models. This way it is able to overcome known limitations of the evaluation methods in the state of the art. In addition, Solon assumes that each DefQA task may require a different evaluation configuration, and it is able to automatically find the best one. The results obtained in our experiments show that Solon performs well with respect to the evaluation methods in the state of the art with the advantage that it is less supervised.

1 Introduction

Typically, the task of Definition Question Answering (DefQA) aims at extracting a definition from text corpora for the target of a definition question. This definition consists of a set of one or more text fragments that contain the most relevant information from the corpus about the question target.

DefQA has received notorious attention in the last years, as show the inclusion of definition questions in question answering competitions such as the Text Retrieval Conference[1] (TREC) and the Cross Language Evaluation Forum[2] (CLEF). However, although the task is becoming more and more relevant, there remains the difficulty of evaluating the performance of definition question answering systems and the quality of the definitions they produce. Available evaluation methods range from manual evaluation to highly supervised ones.

In general, the evaluation methods in the state of the art use the concept of information nugget to score the quality of the definitions provided by a DefQA system. A nugget can be defined as a relevant fact about the target being defined. The reference model of a DefQa task consists of a list of nuggets for each question target in the task. Each nugget is labeled either as vital (required) or okay (optional). The score of a definition produced by a DefQA system depends on the percentage of the vital nuggets from the list that it contains. The score of a DefQA system usually is the average score of the definitions it has produced.

One of the most important DefQA tasks is the question answering track at TREC, that includes a set of definition questions. The official TREC evaluation (Voorhees, 2003; Voorhees, 2004) defines a nugget as a fact for which a human assessor can decide whether it is contained on a system's response or not. The evaluation is performed by manually deciding which nuggets are in the system's response. Table 1 shows an example list of nuggets for a question target extracted from TREC 2004.

Several methods have been proposed to avoid the manual evaluation of TREC. Pourpre (Lin and Demner-Fushman, 2005b) uses a variant of Rouge (Lin and Hovy, 2003) adapted to use information nuggets, as it takes a list of nuggets written by a human assessor as reference model.

Another evaluation method, Nuggeteer (Marton, 2006), goes beyond Pourpre as it also incorporates into the reference model those

[1] http://trec.nist.gov/
[2] http://www.clef-campaign.org/

system responses that a human assessor has already marked as containing a certain nugget.

Nugget #	Type	Description
1	vital	Extensively modified after Challenger accident.
2	vital	Predicted to be used into 2010's.
3	okay	Individual shuttles cost 2 billion.
4	okay	Shuttle payload cost - $10,000 per lb.
5	okay	Shuttles rehabbed with glass cockpits.
6	okay	Shuttle program originated in Nixon years.

Table 1. Nugget list for question target "space shuttles" from TREC 2004.

Finally, in (Lin and Demner-Fushman, 2006) the Pyramid evaluation method (Harnly et al., 2005) is applied to the evaluation of definition questions. The reference model consists of a list of Semantic Content Units (SCUs) built by several assessors. Basically, an SCU is a fact about the question target, and a nugget can be built with one or more SCUs.

All of these methods are designed around the concept of information nugget, which has several drawbacks pointed out in (Hildebrandt et al., 2004; Lin and Demner-Fushman, 2005b). The most important of them is the lack of operational methods neither to create the lists of nuggets nor to classify them as vital or okay. Moreover, only the facts present in the list of nuggets are accepted in the definitions. These factors deeply influence the evaluation of the systems and seriously limit the final quality of the definitions they produce.

In this paper we present a new evaluation framework for the DefQA task called Solon. Solon uses a fully unsupervised evaluation method and it is able to configure itself to achieve a good evaluation procedure adapted to each specific evaluation task.

Solon can use already existing human definitions in order to overcome the problems associated with nugget-based evaluation methods explained above. First, Solon does not need an *a priori* description of which facts are relevant for each target and in which degree, as human definitions tend to contain this information. This means that human definitions can be good models of response to definition questions, and they can be taken as gold standards for DefQA tasks. Second, human definitions can be collected from different sources with minimum supervision efforts.

2 Proposed Evaluation Framework

A DefQA task comprises two main elements. Each definition question asks a target to be defined. Therefore, the first element is a set Q of question targets, where each $q \in Q$ is the target of a definition question. The second element is the set of responses of the automatic systems, A. The response of a system $a \in A$ contains a definition for each $q \in Q$.

Moreover, Solon requires a set M of gold standard reference models. Each model $m \in M$ consists of a human definition for each question target $q \in Q$.

Given all those elements, our framework allows two functionalities:

a) To automatically execute the evaluation procedure on the systems.

b) To automatically reconfigure the evaluation procedure if necessary.

Both functionalities are described in the following sections. The specific DefQA task evaluated in our experiments and the set M of models employed are described in Section 3.

2.1 Evaluation procedure

When gold standard models are used to evaluate automatic systems, it is reasonable that the more similar a system and the models, the better. In order to measure the similarity between human definitions and DefQA systems responses, we need a similarity metric, simple or heterogeneous, that measures the aspects relevant to the specific DefQA task being evaluated. Therefore, system responses with higher similarity values with respect to the models are ranked better than distant ones. The similarity metrics employed in our experiments are described in Section 4.

The evaluation procedure of Solon ranks a system response $a \in A$ according to its similarity to the models in M. As $|M|$ increases the reliability of this approach tends to increase, because a system response is better evaluated if it is compared with a higher number of models. Therefore we need a method that allows to measure the similarity between a system response and a set of models.

The evaluation framework Qarla (Amigó et al., 2005) has been used to evaluate summarization systems with state of the art performance. We use Qarla in our experiments for three main reasons. First, as it has been

pointed out in (Lin and Demner-Fushman, 2005a), DefQA is a task close to automatic summarization, so it is reasonable to use Qarla to evaluate DefQA systems. Second, Qarla uses a probabilistic evaluation procedure that measures how similar is a system response to a set of reference models M using a combination of similarity metrics X. Finally, Qarla is able to evaluate the quality of the evaluation that is performing, as explained in Section 2.2. As a result of the evaluation, Qarla produces the *Queen* measure, $Queen(M,A,X) \in [0,1]^{|A|}$, that is the quality ranking of the automatic systems in A. The quality of a system $a \hat{I} A$ is computed as the probability in $M´M´M$ that, for every metric $x \hat{I} X$, a is closer to a model $m \hat{I} M$ than two other models $m',m'' \hat{I} M$ to each other.

2.2 Configuration of the evaluation

The suitability of a combination of metrics X to evaluate a DefQA task depends on two factors: the elements of the DefQA task (set of question targets, systems responses), and the set of reference models used.

Let S be the set of possible similarity metrics. The space of possible metric combinations, i.e. the solution space is the set of partitions of S, 2^S. As it is computationally unaffordable to exhaustively explore the whole set of combinations, an efficient search algorithm (local beam-search) is used to find the best possible metric combination within a reasonable amount of time. Figure 1 describes the algorithm employed.

The algorithm has three parameters. w is the number of metric combinations that are selected on each iteration. We consider for w a default value of 10. S is the set of possible similarity metrics. The set used in our experiments is described in Section 4. Finally, function h_1 is a search heuristic designed as follows to evaluate the goodness of a metric combination:

$$h_1(X) = King(M, A, X) - \frac{|X|}{100} \qquad (1)$$

The first term, $King(M,A,X) \in [0,1]$, is computed by Qarla as the probability that any reference model $m \hat{I} M$ is better than any automatic system $a \hat{I} A$. This means that Qarla using a metric combination X with a high *King* value ranks reference models better than automatic system responses, i.e. it gives higher scores to better DefQA responses.

The second term, $|X|/100$, measures the length of the metric combination. If two metric combinations obtain approximately the same *King* value, then the shortest combination (in number of metrics) is preferred. The weighting factor of 100 is choosen to reflect the greatest preference of the *King* value in front of the length of the metric combination.

Initially, the algorithm builds a window W_0 with the w individual metrics with highest h_1 value. On iteration i, it generates a set called *succ* of new metric combinations adding individual metrics from S to the combinations $c_j \in W_{i-1}$. The new W_i contains the w combinations with highest h_1 from $W_i \cup succ$. The process is repeated until the combinations in W_i are the same that in W_{i-1}, i.e. no new combination improves the h_1 value. Finally, the best combination from the set W_n is returned.

```
function find_best_metric_combination(w,S,h1)
   W0=best_combinations(w,S,h1)
   possible_successors:=false
   while ¬possible_successors
      succ:=∅
      foreach cj in Wi
         foreach si in S
            if si∉ cj then succ:=succ ∪ add(cj,si)
         endfor
      endfor
      Wi+1:=best_combinations(w,Wi ∪ succ,h1)
      if Wi+1= Wi then possible_successors:=false
   endwhile
   return best_combinations(1,Wi,h1)
endfunction

function best_combinations(w,C,h1)
   return {c1, ..., cw}|ci∈C ∧ ∀ci∈{c1, ..., cw} ∀cj∈C-{c1, ..., cw} h1(ci)≥ h1(cj)
endfunction
```

Figure 1. Search algorithm used to find the best metric combination.

3 DefQA Task and Reference Models

The experiments described in this paper use Solon to evaluate the DefQA systems that participated in TREC 2004 (Voorhees, 2004). This task was the first TREC DefQA task with a stable evaluation method. It contains 64 definition questions and 65 participant systems.

Human definitions can be gathered from Internet, digital libraries, encyclopedias or newspaper collections with minimum supervision. They can also be written *ad hoc* if the evaluation task requires it.

To obtain a set of reference models M in a real evaluation scenario, $|M|$ assessors would collect one definition for each question target $q \hat{I} Q$ in the task. However, in our experiments we have collected $|M|$ definitions from Internet for each $q \hat{I} Q$. Then, the following question arises: how do we distribute the definitions among the models in M? A general way of performing this distribution is to randomly assign the definitions to the models.

Qarla requires a minimum of 3 reference models. In our experiments we use a set of 10 reference models. As we will show in Section 5.3, this is an appropriate number of models for the task

4 Combinations of Similarity Metrics

In order to properly compare two definitions, we need a combination of similarity metrics, each one of them applied to an adequate definition representation. We have experimented with different similarity metrics applied to different representations in order to find the best combination to evaluate the TREC 2004 DefQA task with Solon. The representations and metrics used are described in the following sections.

4.1 Representation of Definitions

Two issues concern the representation of definitions: which attributes are represented and how are they represented.

In Solon, the attributes used to represent definitions are those commonly used in NLP applications: forms, lemmas or WordNet 2.1 synsets (Miller et al., 1990) of the nouns and verbs in the definitions.

A definition is represented as a vector belonging to a vector space A^N, where A is each of the possible values of the attribute used in the representation. In this paper, we use several standard ways to assign a value to each component of a definition vector: binarized values (1 if present, 0 otherwise), the conditional probability of each attribute value with respect to the definition, term frequency (*tf*), inverse document frequency (*idf*) and (*tf·idf*).

4.2 Similarity Metrics

In the experiments described in this paper we have used several text similarity metrics common in different NLP tasks. These metrics are:

- Vector cosinus.
- Rouge-1 (Lin and Hovy, 2003).
- Inverse Jensen-Shannon divergence and the L1-Norm, both used in (Slonim and Tishby, 2000) for document clustering.

Both automatic and manual definitions are relatively long (in number of words). This implies that some attribute values, although non relevant for the definition itself, might influence its evaluation. In order to avoid the noise caused by frequent but non relevant attribute values, we propose a method that acts as a pass-high filter. Let D be the set of all definition models for a given definition question target. Its attribute values would generate a vector space A^N. Then, we define a *relevance space* (*rs*) as a vector space A^K that is a subspace of A^N where:

- Each attribute value of A^N is assigned a *relevance* value that can be calculated through different methods (in our experiments, *tf*, *idf* and *tf·idf*). Relevance values are normalized to the range [0,1].
- The relevance space is characterized by an α value that acts as a minimum relevance threshold: only the attribute values of A^N whose relevance is greater or equal than α are dimensions of the relevance space.

A consequence of the previous conditions is that $K \pounds N$. In particular, if α=0 no attribute value is rejected and therefore it is true that K=N and $A^N = A^K$. A text vector $v \hat{I} A^N$ is projected into a relevance space A^N by a projection function $\Pi: A^N \rightarrow A^K$.

Relevance spaces and the associated projection function are used in a text similarity metric defined by the following formula:

$$mvc(v_1, v_2) = \frac{\Pi(v_1) \cdot \Pi(v_2)}{|v_1| \cdot |v_2|} \qquad (2)$$

We call this similarity function *modified vector cosinus* (*mvc*). The goal of this function is to take into account the original length of the definition vectors. This is important when using relevance spaces to avoid that two long and distant definitions with rather similar projections get a high similarity value. As the *mvc* function makes use of relevance spaces, it depends on the relevance function and the α parameter that define the space. Therefore, it will be referred to as *mvc(relFunc,* α*)*.

Preliminary experiments with the *mvc* function showed that useful values for parameter α are in the interval [0, 0.2]. In any case, the existence of this parameter gives raise to a great number of variants of the same similarity metric.

5 Experiments and Results

In order to test the quality of Solon as evaluation framework for DefQA we have run several experiments that are described in this section. In these experiments we analyze the results by taking into account the following quality measures:

- *King*, described in Section 2.2.
- *Jack*. *Jack(M,A,X)* $\in [0,1]$ is a value produced by Qarla that measures if *Queen* and *King* results are reliable for the sets *M*, *A* and *X* employed. It is defined as the probability that for every $m \hat{I} M$ be a couple of automatic systems that are closer themselves than to *m*. Intuitively, it tends to 0 if *M* and *X* are unable to distinguish one $a \hat{I} A$ from the other and it tends to 1 otherwise.
- *LOOprec* is the precision of the framework in a leave-one-out test that puts a model among the systems responses and succeeds if Solon ranks it as the best system. Its objective is to have another measure of the quality of the evaluation, as a better system is ranked better.
- *Correl*. The evaluation performed by a framework is usually considered good if it is close to an evaluation made by human assessors. Therefore, we also include as a quality measure the correlation of the evaluation performed by Solon with the official TREC evaluation, expressed with the following formula:

$$Correl(M, A, X) = R^2(TREC(A), Queen(M, A, X)) \quad (3)$$

  where R^2 is Pearson's R^2 correlation, *TREC(A)* is the official TREC evaluation for the set of automatic systems *A* and *Queen(M,A,X)* is the ranking of the systems produced by Solon.
- *StdDev*. In Section 3 we said that the reference models used in the evaluation are generated by randomly distributing the definitions for each question target among them. In order to see if Solon produces an robust evaluation regardless of this distribution, we generated 10 different sets of reference models and run all the experiments with each of them. The results given for the quality measures listed above are their means for these 10 runs. Therefore we have new quality measures, the standard deviations of *King*, *Jack*, *LOOprec* and *Correl* measures. In all cases this standard deviation was around 1%. This is a sign of the robustness of Solon with respect to distribution of definitions among reference models.

The goals, configurations and results of each experiment are described next.

5.1 Best Metric Combination Search

The goal of the first experiment is to find the best metric combination to evaluate DefQA systems. The search algorithm described in Section 2.2 is applied to the similarity metrics listed in Section 4.2.

The results of the experiment, summarized in Table 2, are explained next:

- X_1 is the best metric combination found using heuristic h_1. It is a combination of modified vector cosinus applied to both lemmas and synsets.
- X_2 is the best individual metric. Its α is lower (0.04) than in X_1 (0.09 for lemmas). This is probably due to the fact that, while X_1 has also synset information to perform the evaluation, X_2 does not and therefore it requires more information in its relevance space (a lower α threshold). As was explained in Section 4.2, lower α values imply more terms in the relevance space.

Metric comb.	$H_1(X)$	King	Jack	LOO prec	Correl
X_1	**0.593**	**0.613**	0.978	**1.000**	0.862
X_2	0.575	0.585	0.990	**1.000**	0.762
X_3	0.531	0.541	**0.994**	**1.000**	**0.869**
X_4	0.357	0.367	0.985	0.140	0.774

X_1={mvc(tf,0.09)_bin_lem, mvc(tfidf,0.05)_bin_syn}
X_2={mvc(tfidf,0.04)_bin_lem}
X_3={cos_bin_lem}
X_4={rouge1_bin_form}

Table 2. Summary of results of best metric combination search experiment.

- X_3 is the best metric combination that does not make use of relevance space metrics, in this case only with one metric, the vector cosinus applied to binarized lemmas.
- X_4 is Rouge-1 applied to binarized word forms. It is taken as a baseline evaluation method because DefQA evaluation methods listed in Section 1 can not be used with reference models built from human definitions.

The metric combinations with highest heuristic values use lemmas and synsets as text attributes, but not word forms. This implies that the quality of a definition depends more on its semantic than on its form.

In (Amigó et al., 2005), Qarla is used to perform state of art evaluation of summarization systems with a *King* value of 0.47. In this experiment, the best metric combination (X_1) achieves a *King* value of 0.613. This result confirms that Solon is a suitable framework to perform state of art evaluation of DefQA systems.

Jack values obtained by the metric combinations listed in Table 2 are all above 0.97. This means that Solon is able to distinguish among the systems being evaluated, judging each one according to its features and its quality.

All the metric combinations listed except Rouge1 are able to distinguish a reference model from the automatic systems' responses (*LOOprec* column). This confirms the ability of the framework to rank better systems higher. A standard evaluation method as Rouge1 only detected the reference model in 14% of the tries.

The correlation with TREC official evaluation (*Correl* column) is analyzed in Section 5.2.

5.2 Correlation with TREC Evaluation

The correlation with TREC official evaluation obtained by the best metric combination (X_1) is R^2=0.862 (see Table 2). This value is lower than the correlation values of other DefQA evaluation methods, shown in Table 4.

This lower correlation result has two main causes. First, the search heuristic used to obtain the metric combination X_1 is not designed at all to achieve a high correlation with TREC or other evaluation methods. Second, the reference models used by Solon are clearly different from those used by TREC or the other evaluation methods. While Solon uses human written definitions, the other methods use lists of information nuggets or similar to perform the evaluation.

As we previously said in Section 5, a quality measure of an evaluation framework is its degree of correlation with a manual evaluation method. It is possible to instruct Solon to achieve a higher correlation with TREC official evaluation by replacing the heuristic h_1 with the following:

$$h_2(X) = Correl(M, A, X) - \frac{|X|}{100} \qquad (4)$$

Where *Correl* is the correlation function defined in Formula 3. The second term of the formula penalizes longer metric combinations. In other words, this heuristic is designed to maximize the correlation with TREC official evaluation while keeping a metric combination with as few metrics as possible.

Metric combination	$H_2(X)$	Correl(M,A,X)
X_5	**0.908**	**0.918**
X_1	0.842	0.862
X_4	0.764	0.774

X_5={mvc(tf,0.12)_bin_syn}

Table 3. Extract of results for highest correlation search experiment.

Table 3 presents the most relevant results of this experiment. The metric combination with most correlated with TREC evaluation is X_5, that uses the modified vector cosinus on binarized synsets. TREC assessors evaluate systems responses by their semantics, deciding whether an information nugget is included in the response regardless of the form in which it is expressed. It makes sense that X_5 is the metric with highest correlation, as it also uses semantics (synsets) to evaluate system responses.

Table 4 compares the correlation values obtained by Solon with those from the other automatic evaluation methods, both as R^2 and as Kendall's τ.

Evaluation method	R^2	Kendall's t
Pourpre	0.929	0.833
Nuggeteer	**0.982**	0.898
Nugget Pyramids	N/A	**0.943**
X_1	0.862	0.763
X_4	0.774	0.695
X_5	0.918	0.825

Table 4. Correlation with TREC 2004 official evaluation of the evaluation methods available.

The correlation with official TREC evaluation is not as high as that of the other evaluation methods. However, we believe that the correlation values achieved are good, considering that, while the other methods start with the same or a very similar set of reference models of TREC, Solon takes a rather different approach. This also demonstrates that it is a flexible evaluation framework that can be oriented to different evaluation goals.

5.3 Influence of the Number of Models

In the previous experiments we have used a set of reference models M with $|M|=10$ to perform the evaluation. However, a question arises: does that number of reference models allow to achieve a good evaluation?

In order to answer that question, we have run the evaluation framework with different sizes of M, from 3 (the minimum required by Qarla) up to 10 reference models.

All the quality measures showed in Figure 2 tend to stabilize with increasing values of $|M|$. In all runs, *LOOprec* equals 1, so there is a perfect distinction between automatic systems and reference models[3].

The results of this experiment allow to draw several conclusions. First, Solon exhibits an stable behaviour with respect to the number of reference models used. Its quality measures improve with more reference models. Second, the experiment confirms as adequate the initial election of $|M|=10$, as with that number of reference models the quality measures and therefore the evaluation is stable. Finally, the results evidence that with more reference models the evaluation is better. Depending on the specific evaluation scenario, however, it is possible to obtain a similar evaluation with a given $|M|$ than with more reference models.

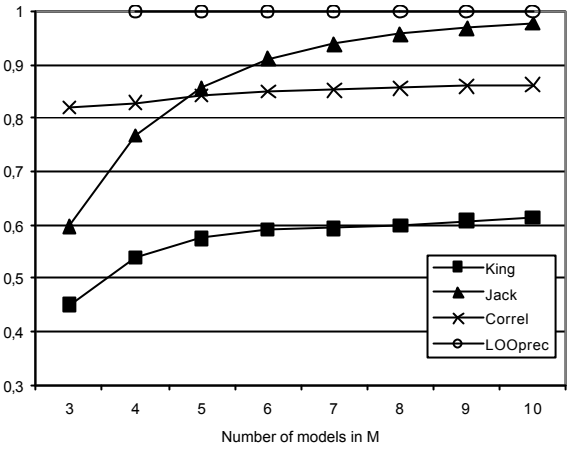

Figure 2. Quality measures obtained by the best metric combination (X_1) varying the number of reference models (size of M).

5.4 Influence of the Incompleteness of Models

The reference models used in the previous experiments where all complete, i.e. they contained a definition for every question target q in the set of definition questions Q of the evaluation task.

In a real evaluation scenario there is a team N of human assessors. Each one of them creates a reference model by collecting or writing definitions for the question targets in Q. We define the *workload* of a human assessor $n \hat{I} N$ as:

$$workload(n) = \frac{|m_n|}{|Q|} \qquad (5)$$

where m_n is the reference model created by the assessor n, and its size is the number of question targets it has a definition for. In other words, the workload of a human assessor is the percentage of questions he must define to build a reference model.

In order to study the influence of this factor on the evaluation and see if it can be reduced, we have run the evaluation framework varying the workload required to build the reference models in M from 0.2 to 1.0.

The results of the experiment, represented in Figure 3, show that all the quality measures tend to stabilize when the workload tends to 1. The conclusions of this experiment are the following. First, higher workload values produce higher quality evaluations. In any case, it is preferable

[3] The *LOOprec* test can not be run with $|M|=3$ because if we leave one model out, only 2 remain and Qarla requires a minimum of 3.

to use a *workload* equal to 1 to achieve the best evaluation, as was done on the previous experiments. However, depending on the specific evaluation task it is possible that a *workload* lesser than 1 allows to obtain an evaluation of similar quality than the evaluation using *workload*=1. Second, Solon behaves in a consistent way, improving its evaluation with increasing workload values.

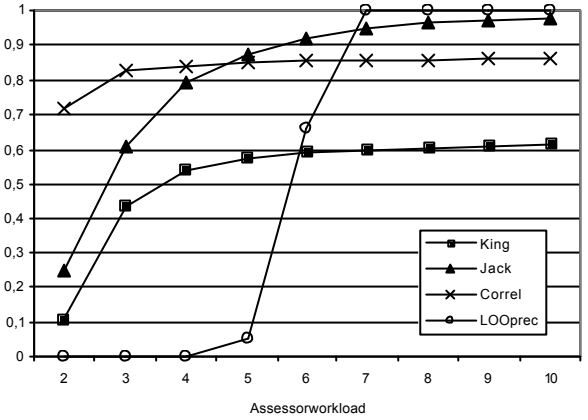

Figure 3. Measures obtained by the best metric combination (X_1) varying the number of definitions for each target.

6 Conclusions

In this paper we present a new evaluation framework for DefQA tasks, Solon, that uses gold standard models to perform the evaluation. It has two functionalities: the automatic evaluation of DefQA systems and the automatic search of an evaluation procedure as much suitable as possible for the task.

For the evaluation of the systems Solon requires a method to compare systems and models using a combination of similarity metrics. For searching the appropriated evaluation procedure, Solon uses a local beam-search that finds a good similarity metric combination from a set of them.

In our experiments we have used Qarla as comparison method and a set of similarity metrics, most of them widely used in different NLP tasks. Nonetheless, Solon is open to the use of other metrics or comparison methods.

The results of our experiments show that our framework is highly correlated with TREC official evaluation and, although this correlation is slightly lower than those achieved by other methods, we think that ours requires less human supervision.

In addition, on the contrary of other evaluation approaches, our framework automatically adapts its evaluation procedure to each specific DefQA task.

References

Enrique Amigó, Julio Gonzalo, Anselmo Peñas and Felisa Verdejo. 2005. QARLA: a framework for the evaluation of text summarization systems. *Proc. of ACL 2005*.

Aaron Harnly, Ani Nenkova, Rebecca Passonneau and Owen Rambow. 2005. Automation of summary evaluation by the Pyramid Method. *Proc. of RANLP 2005*.

Wesley Hildebrandt, Boris Katz, and Jimmy Lin. 2004. Answering Definition Questions with Multiple Knowledge Sources. *Proc. of HLT/NAACL 2004*.

Jimmy Lin and Dina Demner-Fushman. 2005a. Evaluating Summaries and Answers: Two Sides of the Same Coin? *Proc. of ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*.

Jimmy Lin and Dina Demner-Fushman. 2005b. Automatically Evaluating Answers to Definition Questions. *Proc. of HLT/EMNLP 2005*.

Jimmy Lin and Dina Demner-Fushman. 2006. Will Pyramids Built of Nuggets Topple Over? *Proc. of HLT/NAACL 2006*.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic Evaluation of Summaries using n-gram Co-ocurrence Statistics. *Proc. of HLT/NAACL 2003*.

George A. Miller, Richard Beckwith, Christine Fellbaum, Derek Gross and Katherine Miller. 1990. Five Papers on WordNet. *International Journal of Lexicology, 3(4), 1990*.

Noam Slonim and Naftali Tishby. 2000. Document Clustering using Word Clusters via the Information Bottleneck Method. *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information retrieval*.

Ellen M. Voorhees. 2003. Evaluating Answers to Definition Questions. *Proc. of HLT/NAACL 2003*.

Ellen M. Voorhees. 2004. Overview of the TREC 2004 Question Answering Track. *Proc. of TREC 2004*.