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

Samir Kanaan and Jordi Turmo
Technical University of Catalunya
Barcelona, Spain
{skanaan,turmo}@lsi.upc.edu

Abstract
This paper presents a weakly 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 allows the framework to overcome known limitations of the evaluation methods in the state of the art with the advantage that it is less supervised. In addition, Solon adapts its configuration for each specific DefQA task, thus rendering a good evaluation procedure. The results obtained in our experiments show that Solon is able to detect the best systems and to score them accordingly, with state of the art performance.

1 Introduction
Typically, the goal of a Definition Question Answering (DefQA) task is to extract definitions from plain text. A typical DefQA task comprises two elements: a set of definition questions and a text corpus.

Definition questions are usually of the form What/Who is <target>? where <target> is the question target, i.e. the term or entity to be defined, as for example “What are space shuttles?” or “Who is BB King?”. The question target can be a common name or an entity of any kind: person, organization, event, etc. In this paper, we name \( Q \) to the set of question targets of a DefQA task.

The output of system dealing with a DefQA task usually consists of a set of one or more textual fragments extracted from the text corpus.

The fragments extracted for a given target are supposed to contain the most relevant facts about it that could be found in the corpus. The following fragments are an example of a definition generated by a DefQA system for target “space shuttles” (with document identifiers in square brackets):

[APW19990915.0233] The space agency was unable to move the $2 billion-a-piece shuttles to a safer location earlier in the week because it does not have enough modified jumbo jets to transport all four spaceships. [NYT20000907.0214] The shuttles flying today have progressed since the Columbia orbiter first took to space in 1981.

DefQA has received notorious attention in the last years, as show the addition of definition questions to several competitions such as the Text Retrieval Conference\(^1\) (TREC), the Cross Language Evaluation Forum\(^2\) (CLEF) and the Document Understanding Conference 2004 \(^3\) (DUC 04). In turn, this focus on DefQA arises from the fact that users are interested in definitions. Nowadays web search engines receive an increasing number of definition queries.

However, the development of DefQA systems faces a serious problem: it is very difficult to evaluate the quality of the definitions they produce and the evaluation methods in the state of the art present several drawbacks.

In this paper we present a weakly supervised evaluation framework for DefQA tasks called Solon. It uses human definitions as gold standard.

---
\(^1\) http://trec.nist.gov/
\(^2\) http://www.clef-campaign.org/
\(^3\) http://duc.nist.gov/
reference models in order to overcome the drawbacks of the existing evaluation methods.

The evaluation methods in the state of the art are described in Section 2. Section 3 presents the general structure of the evaluation framework Solon. Section 4 describes the configuration of the experiments carried out to evaluate Solon and Section 5 presents their results. Section 6 concludes the paper.

2 Related Work

The existing evaluation methods are all based on the concept of information nugget, or simply nugget, to score the quality of the definitions produced by a DefQA system. In general, a nugget is a relevant fact about the question target. The reference model of a DefQA consists in a list of nuggets for each question target \( q \in Q \). A perfect definition is expected to contain text fragments supporting all of these nuggets. Despite being based on the same concept, those evaluation methods range from manual ones to highly supervised ones.

One of the most important DefQA tasks in the state of the art is defined in the question answering track at TREC. The official TREC evaluation procedure (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 output or not. The evaluation is performed in two steps: first, human assessors manually build a list of nuggets for each question target and label each nugget as either vital (required) or okay (optional). Second, the assessors manually decide which nuggets are present in the output of the systems. Table 1 shows an example list of nuggets for question target “space shuttles” from TREC 2004 DefQA task. The score of a definition at TREC is computed as the f-score between vital nugget recall (percentage of vital nuggets contained in the definition) and an approximation to nugget precision (total number of nuggets in the definition versus definition length). The score of a DefQA system is the median of the scores of the definitions it has produced.

Several automatic evaluation 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.

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

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

Another evaluation method, Nuggeteer (Marton and Radul, 2006), goes beyond Pourpre as it also incorporates into the reference model those system responses that a human assessor has already marked as containing a certain nugget.

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 the evaluation methods presented above are designed around the concept of information nuggets, which has several drawbacks (Hildebrandt et al., 2004; Lin and Demner-Fushman, 2005b). The most important drawback of DefQA evaluation with nuggets is the lack of operational methods to create the list of nuggets corresponding to a question target. Two different assessors would probably build different lists of nuggets for a given question target. This is a serious problem, as only the facts in the list of nuggets are taken into consideration when evaluating a definition. Moreover, there is no method to ensure that a list of nuggets for a given target is complete.

A complementary problem is the overlapping among the nuggets assigned to a question target, as sometimes happens that two or more nuggets in a list overlap partially.

Moreover, the existing evaluation methods make a vital/okay distinction among nuggets. But there are no systematic criteria to classify a nugget as either vital or okay, and this classification deeply determines the evaluation procedure and which information will be used or discarded when evaluating the definitions.
In conclusion, the creation of lists of nuggets requires much human effort and there is no systematic method to build them.

3 Our Proposal

The purpose of our evaluation framework is to evaluate the systems that participate in a DefQA task. Instead of information nuggets, it uses human definitions as gold standard reference models in order to overcome the limitations associated with nuggets. First, Solon does not limit the evaluation of definitions to the facts in a list, but takes into consideration every fact that a human has included in a definition. Second, the nugget overlapping problem does not hold using complete definitions instead of lists of nuggets. Finally, the problematic vital/okay distinction is avoided using several definitions for each question target, as the most frequent facts are deemed as most important.

Solon can use already existing human definitions or definitions written ad hoc for the DefQA task being evaluated. The following is an extract of a human definition for target “space shuttles” taken from http://en.wikipedia.org:

NASA's Space Shuttle, officially called Space Transportation System (STS), is the United States government's current manned launch vehicle. The winged shuttle orbiter is launched vertically, usually carrying five to seven astronauts (...) and up to 22,700 kg (50,000 lb) of payload into low earth orbit. When its mission is complete, it re-enters the earth’s atmosphere and makes an unpowered horizontal landing. (...)

Our framework offers two functionalities: 1) as other evaluation methods, to automatically execute an evaluation procedure on the systems participating in a DefQA task. 2) Unlike the other evaluation methods, Solon automatically configures the evaluation procedure in order to obtain the best possible configuration to evaluate a specific DefQA task.

When Solon performs the evaluation of the systems participating in a DefQA task, it uses the following inputs:

- A set $M$ of gold standard reference models. Each model $m \in M$ contains one human definition for each $q \in Q$.

As a result of the evaluation, Solon produces two outputs:

- The score assigned to each of the systems in $A$.
- A measure of the quality of the evaluation performed.

The evaluation of the DefQA task using Solon comprises two steps. First, Solon finds the best evaluation configuration for $M$ and $A$. The strategy of choosing the configuration of the evaluation a posteriori has been successfully used in (Amigó et al., 2005) for summarization systems. Solon produces a quality value that allows to know the quality of the evaluation that will be obtained.

In second place Solon applies the evaluation procedure with the configuration selected to score the systems in $A$ using $M$ as reference models.

3.1 Architecture of the Evaluation Framework

The architecture of Solon is shown in Figure 1. Besides its inputs and outputs, explained above, Solon is structured in three modules:

- A text similarity module containing a set of text similarity functions. Each of these functions receives a pair of definitions and returns their similarity according to some text similarity metric. The configuration of Solon for a specific DefQA task is a set of similarity metrics, referred to as $X$ in this paper. The specific similarity metrics used in our experiments are listed in Section 4.3.

- An evaluation module responsible of evaluating the systems with respect to the models using a set of similarity metrics $X$. This module produces both the scores of the DefQA systems and a measure of the quality of the evaluation performed, $Quality(M, A, X)$.

- A configuration module that explores different sets of metrics and finds the most suitable set to evaluate a specific DefQA task according to $Quality(M, A, X)$.
The configuration module selects a configuration (a set of metrics), instructs the similarity module to compute the similarities of the systems and the models according to these metrics, and feeds these similarities to the evaluation module to find the quality of the configuration. The configuration module then takes this quality value from the evaluation module in order to select the most suitable configuration. When it finds the best suited configuration, Solon outputs the scores of the systems according to that configuration and the quality value associated with it. The internals of the evaluation and configuration modules are explained next.

**Evaluation Module**

Solon uses an evaluation module with the following properties:

- It scores a set of DefQA system outputs $A$ according to their similarity to a set of gold standard models $M$.
- This similarity can be computed using a set of similarity metrics, disregarding scale issues among similarity metrics.
- It produces a $\text{Quality}(M, A, X)$ value that measures the quality of the evaluation performed.

The evaluation module that we use in the experiments presented in this paper is Qarla (Amigó et al., 2005) for two main reasons. First, it has been successfully used to evaluate summarization systems with state of the art performance. 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 exhibits all the properties listed above.

Qarla is an evaluation framework that, given a set $M$ of gold standard models, a set $A$ of system outputs to be evaluated, and a set of similarity metrics $X$, produces a set of probabilistic evaluation measures. From this set, Solon uses the King and Queen measures:

- $\text{Queen}(M, a, X) \in [0, 1]$ is the score of an automatic system $a \in A$. This score is computed as the probability that the similarity between $a$ and a model $m \in M$ is greater or equal than the similarity between any pair of models $m', m'' \in M$ for all the similarity metrics $x \in X$. In other words, it assigns higher scores to the systems more similar to the models.
- $\text{King}(M, A, X) \in [0, 1]$ is the probability that any reference model $m \in M$ be better than any automatic system $a \in A$ using the metric set $X$. A high $\text{King}$ value guarantees 1) that the models in $M$ are better ranked than the automatic systems, following the idea that human models are better that automatic ones, and 2) that the

![Figure 1. Architecture of the Evaluation Framework Solon](image-url)
similarity metrics in $X$ capture the features common to the models in $M$ and therefore they can be used to properly evaluate the $A$ with respect to the common features of the models in $M$. As a consequence, King measures the suitability of $X$ to evaluate $A$ with respect to $M$. Consequently, Solon uses the Queen measure as the score of each automatic system ($System Scores$ output of Solon) and the King value as a measure of the quality of the evaluation performed ($Quality(M, A, X)$ output). In turn, this quality value is used to find the best metric set for a specific evaluation task, as is described in the following section.

A consequence of the definition of Queen is that Qarla requires a minimum of three models, although a greater number of models improves its performance. In the context of summarization, it has been applied using 9 models with state of the art performance (Amigó et al., 2005). It has also been used to evaluate “Who-is” summaries, as described in (Amigó et al., 2004).

**Configuration Module**

The quality of a set of similarity metrics $X$ to evaluate the set of DefQA systems $A$ using the models $M$ ($Quality(M, A, X)$) depends on both $A$ and $M$. Such quality value is computed by Solon as $King(M, A, X)$.

In order to perform the best possible evaluation, Solon has to find the most suitable $X$ given $A$ and $M$. Being $S$ the set of available similarity metrics, $X$ is a subset of $S$. In order to find the most suitable $X$, Solon has to explore the space of possible metric sets, i.e. the solution space will be the set of partitions of $S$, $2^S$. As it is computationally unaffordable to exhaustively explore the whole solution space, Solon uses an efficient search algorithm (local beam-search) to find the best possible metric set within a reasonable amount of time.

The algorithm employed, shown in Figure 2, receives three parameters: 1) $w$ is the number of metric sets that are selected on each iteration. 2) $S$ is the set of similarity metrics tried. 3) Finally, function $h$ is a search heuristic designed to evaluate the goodness of a metric combination.

In our experiments we use $w=10$, the set of similarity metrics $S$ listed in Section 4.3, and the following heuristic for $h$:

$$h'(X) = King(M, A, X) - \frac{|X|}{1000}$$  \hspace{1cm} (1)

The purpose of the second term is to choose the metric set with fewer elements when comparing two sets with a King value approximately equal. We consider that a difference of less than 0.1% is not significant.

The algorithm keeps a window of size $w$ with the metric sets with highest $h'$ value. This window initially contains the best individual metrics. On each iteration, the algorithm creates new sets adding individual metrics to the sets in the window. The window is updated taking into account these new sets. The algorithm stops when the window does not get updated on an iteration, and yields the metric set with highest $h'$.

**4 Experiments**

The experiments described in this paper use

```python
function find_best_metric_combination(w,S,h)
    W_i=best_combinations(w,S,h)
    possible_successors:=false
    while ¬possible_successors
        succ:=∅
        foreach c_j in W_i
            foreach s_i in S
                if s_i ∉ c_j then succ:=succ ∪ add(c_j,s_i)
            endfor
        endfor
        W_{i+1}:=best_combinations(w,W_i ∪ succ,h)
        if W_{i+1}= W_i then possible_successors:=false
    endwhile
    return best_combinations(1,W_i,h)
endfunction

function best_combinations(w,C,h)
    return {c_1, ..., c_w}|c_j∈C ∧ ∀c_k∈{c_1, ..., c_w} ∀c_l∈C−{c_1, ..., c_w} h(c_k) ≥ h(c_l)
endfunction
```

Figure 2. Search algorithm used to find the best metric combination.
Solon to evaluate the DefQA systems that participated in TREC 2004 QA Track (Voorhees, 2004). This task was the first TREC DefQA task with a stable evaluation method. It contains 64 definition questions and 65 participant systems. We have chosen this task for our experiments because it has the greatest number of questions and participants and it contains question targets of several types besides human ones.

### 4.1 Reference Models

Solon requires a set of reference models $M$ to perform the evaluation. In our case, three assessors have gathered definitions from the web so as to build 10 reference models, each with one definition for each of the 64 question targets. A general way of distributing the definitions is to do it randomly. To study the influence of the random distribution of definitions we have generated ten random distributions and have run the experiments with all of them. The results presented in Section 5 are the means of the results of these ten runs.

### 4.2 Evaluation Measures

The goal of the experiments is to evaluate the evaluation framework, therefore we need to measure the goodness of the evaluation obtained. The measures employed are:

- $\text{King}(M, A, X)$
- $\text{LOOP}$: leave-one-out precision. Its purpose is to test if Solon is able to assign highest scores to the best systems. We take one model $m_{out}$ out of $M$ and insert it into $A$. Then Solon, with $A \cup \{m_{out}\}$ as systems and $M - \{m_{out}\}$ as models, finds the best configuration and performs the evaluation. $\text{LOOP}$ measures the percentage of times that Solon scores $m_{out}$ as the best system, which ideally should be 1, meaning that it always finds a human (better) definition among a set of automatic definitions. This process is applied to all the models in $M$, and Solon uses a different metric set for each model left out.
- $R^2$: $R^2$ correlation of the scores produced by Solon with the official TREC 2004 scores. Although it is not the goal of Solon to reproduce the evaluation of TREC, we have included this correlation for informative purposes.

### 4.3 Similarity Functions

The similarity module employed in our experiments implement metrics well known in NLP:

- Vector cosinus applied to binarized documents, term frequency ($tf$), inverse document frequency ($idf$) and $tfidf$.
- Intersection, Information Radius, Dice and Jaccard, taken from (Manning and Schütze, 1999).
- Inverse Jensen-Shannon divergence and L1-Norm, both used in (Slonim and Tishby, 2000) for document clustering.
- Rouge-1, 2 and 3 (Lin and Hovy, 2003).

Each of the previous similarity metrics has been computed using used either word forms, lemmas and WordNet 2.1 synsets (Miller et al., 1990) of verbs and nouns as attributes.

### 4.4 Definition of the Experiments

In order to evaluate Solon and study its behaviour, we have run three experiments whose results are presented in Section 5.

First we have applied Solon to the evaluation of the DefQA systems in TREC 2004 QA Track. This experiment allows to test Solon in a real evaluation task.

In second place we study the influence of the number of models in $M$ on the behaviour of Solon. Although we have chosen 10 models for the previous experiment, it is important to study if this number of models is appropriate or not. In this experiment we vary the number of models in $M$ from 3 (the minimum required by Qarla) to 10 and analyze the evaluation obtained.

In the third experiment we study the influence of the incompleteness of the models. The reference models used in the previous experiments are all complete, i.e. they contain 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 $n \in N$ as:

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

(2)

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 varying the workload required to build the reference models in $M$ from 0.3 to 1.0.

5 Results

Recalling from Section 4.1, the results presented in this section are the means of 10 different distributions of definitions among models. The standard deviations are in all cases below 1%, and therefore they are not specified.

5.1 Evaluation of TREC 2004 with Solon

Table 2 summarizes the results obtained, with $X_1$ being the metric set selected by Solon. $X_2$ and $X_3$ are the two Rouge configurations with highest King value and are presented in the table as reference configurations.

<table>
<thead>
<tr>
<th>Metric set</th>
<th>$h^1$</th>
<th>King</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>0.592</td>
<td>0.597</td>
</tr>
<tr>
<td>$X_2$</td>
<td>0.413</td>
<td>0.414</td>
</tr>
<tr>
<td>$X_3$</td>
<td>0.471</td>
<td>0.472</td>
</tr>
</tbody>
</table>

$X_1=\{\text{cosbin\_form, cosbin\_synset, cosidf\_lemma, cosidf\_synset, intersection\_lemma}\}$

$X_2=\{\text{rouge1\_form}\}$

$X_3=\{\text{rouge2\_form}\}$

Table 2. Summary of results using Solon to evaluate TREC 2004. The metrics are expressed as metricName_attribute.

The metric set selected by Solon achieves a high King value compared to the baseline configurations. This means that $X_1$ captures the common features of the models when evaluating the systems.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>LOOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solon</td>
<td>0.93</td>
</tr>
<tr>
<td>rouge1_form</td>
<td>0.51</td>
</tr>
<tr>
<td>rouge2_form</td>
<td>0.63</td>
</tr>
</tbody>
</table>

Table 3. LOOP results for both the configuration proposed by Solon and the two baseline configurations.

The LOOP results shown in Table 3 indicate that using the metric sets selected by Solon it is possible to score better a human model than all of the 63 system outputs in 93% of the cases. This clearly indicates that, in most cases, Solon is able to find the best system from a set. The performance of the baseline configurations is clearly below. Recalling from Section 4.2, the baseline systems tested are an approximation to some of the evaluation methods in the state of the art. Therefore the performance of these methods is reasonably expected to be similar to the performance of the baseline systems, which is clearly below the performance achieved by Solon.

In conclusion, the adaptability of Solon improves the quality of the evaluation procedure.

Solon evaluates the outputs of DefQA systems according to their similarity with human definition models. This is a very different approach to that of the other evaluation methods, that evaluate DefQA systems searching for information nuggets. As a consequence, as shown in Table 4 the scores assigned by Solon are less similar to the official TREC scores than those assigned by nugget-based methods.

<table>
<thead>
<tr>
<th>Evaluation method</th>
<th>$R^2$</th>
<th>Kendall’s $\tau$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pourpre</td>
<td>0.929</td>
<td>0.833</td>
</tr>
<tr>
<td>Nuggeteer</td>
<td>0.982</td>
<td>0.898</td>
</tr>
<tr>
<td>Nugget Pyramids</td>
<td>N/A</td>
<td>0.943</td>
</tr>
<tr>
<td>$X_1$</td>
<td>0.894</td>
<td>0.787</td>
</tr>
<tr>
<td>$X_2$</td>
<td>0.761</td>
<td>0.694</td>
</tr>
<tr>
<td>$X_3$</td>
<td>0.787</td>
<td>0.674</td>
</tr>
</tbody>
</table>

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

5.2 Influence of the Number of Models

In this experiment we have used the metric set $X_1$ obtained in the previous experiment to study the influence of the number of models in $M$. The
evaluation measures, showed in Figure 3, improve with increasing values of $|M|$, although they also tend to stabilize\(^4\).

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 are stable. Finally, the results evidence that with more reference models the evaluation is better. Depending on the specific evaluation scenario, however, a certain value of $|M|$ may render an evaluation similar to what would be obtained using more reference models.

The conclusions of this experiment are the following: first, higher workload values produce higher quality evaluations. In any case, it is preferable 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 with a workload equal to 1. Second, Solon behaves in a consistent way, improving its evaluation with increasing workload values.

5.3 Influence of the Workload of Models

In this experiment we have used the metric set $X_1$ previously obtained to study the influence of the workload required to build the models in $M$. The results of the experiment, represented in Figure 4\(^5\), 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 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 with a workload equal to 1. Second, Solon behaves in a consistent way, improving its evaluation with increasing workload values.

6 Conclusions

Definition question answering, or DefQA, is a growing field in the NLP area. The existing evaluation methods for DefQA require high supervision and are all based on the concept of information nugget, which has several drawbacks and limitations.

In this paper we present a new evaluation framework for DefQA tasks called Solon that uses gold standard models to perform the evaluation. This evaluation comprises two steps: first, Solon finds the best suited configuration for the specific task; second, it applies this configuration to evaluate and score the outputs of the participant systems.

In the experiments presented in this paper we have employed several similarity metrics commonly used in NLP, and Qarla as evaluation module.

---

\(^4\) The LOOP test is run with the best configuration obtained by Solon disregarding the model left out, i.e. it is not run with $X_1$. The 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.

\(^5\) As in Figure 3, the LOOP test is run with the best configuration for the set of models $M-\{m_{\text{out}}\}$.
specific DefQA task to be evaluated. Third, Solon is open to use different similarity metrics or evaluation methods.

The results of the experiments show that Solon is able to rank better systems in the first position in 93% of the cases, achieving state of the art performance in the task.

Acknowledgements

We wish to thank Enrique Amigó for providing us the Qarla software along with his valuable suggestions. This work has been partially supported by the European Union, project CHIL (IP 506909).

References


