Retrieving Answers to Definition Questions

Alejandro Figueroa
Deutsches Forschungszentrum für Künstliche Intelligenz - DFKI
Stuhlsatzenhausweg 3, D - 66123, Saarbrücken, Germany
figueroa@dfki.de

ABSTRACT

WebQA is a Web Question Answering System [1] which is aimed at discovering answers to natural language questions on the web. One of its major components is the module that answers definition questions. A crucial aspect of this module is that it searches for answers by means of a query rewriting strategy, which considerably boosts the recall of descriptive utterances. This study compares three different search strategies, and additionally, it deals at greater length with the challenges posed by the assessment of web-based definition Question Answering Systems.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Content Search and Analysis; I.2.7 [**Artificial Intelligence**]: Natural Language Processing

General Terms

Algorithms, Experimentation

Keywords

Web Question Answering, Definition Questions, Web Mining, Definition Question Answering

1. INTRODUCTION

WebQA is part of sustained efforts to implement a system which extracts answers to open-domain factoid[4] and definition[6], as well as list questions[5] exclusively from the brief descriptions (web snippets) returned by commercial search engines, like Google and MSN Search as well as Yahoo.

The reason to use web snippets as an answer source is four-fold: (a) they are computed at high speed by current commercials search engines, and therefore provide a quick and contextualised response, (b) to take advantage of the current power of indexing of vanguard search engines, (c) to the user, web snippets are the first view of the response, thus highlighting answers would make them more informative, and (d) to avoid, or at least lessen, the retrieval and costly processing of a wealth of documents. In particular, web snippets have proven to be promising for answering difficult queries like definitions questions (such as "*Who is George Bush?*", "*What are fractals?*" or "*What is AI?*"). This sort of query is particularly important, because 27% of the questions of real user logs are a request for a definition. In order to satisfactorily answer definition questions, Question Answering Systems (QAS) must take answers from several documents and afterwards, discriminate senses, merge answers, remove redundancy, and eventually generate a final output for the user. This study focus its attention on definition questions, especially on the first step: the search or retrieval of definition answers.

The roadmap of this paper is as follows: section 2 deals at greater length with the related work, and section 3 describes WebQA in brief. Section 4 proposes a new search strategy for WebQA. Accordingly, section 5 shows results, and eventually, section 6 draws conclusions.

2. RELATED WORK

QAS are usually assessed in the context of the Question Answering track of the Text REtrieval Conference (TREC). In TREC, the target collection is the AQUAINT corpus. Broadly speaking, in order to successfully discover right answers to definition questions, definition QAS align some syntactic patterns with sentences. The probability of matching sentences, therefore, increases as long as the target collection grows in size, and consequently, the performance noticeably improves [10]. Afterwards, the most promising matched sentences are normally selected by weighting the following three criteria [3, 8, 9, 10]: (a) the accuracy of the pattern that signals the corresponding descriptive sentence, (b) frequencies of words within matched sentences, given that high frequent terms are very likely to belong to descriptions, and (c) frequencies of words that co-occur with the target concept (a. k. a. the *definiendum*), given that they are likely to express its definition facets [3]. These ranking criteria proved to work satisfactorily for a set of 146 questions and their corresponding 600 top-ranked full-documents retrieved from the web [10].

In addition, QAS make use of several external resources of information that supply definition nuggets. QAS then identify descriptive phrases by projecting these external nuggets into the target corpus. In this way, they also filter out some misleading and spurious nuggets taken from these external sources. In the jargon of definition questions, a nugget is a piece of relevant or factual information about the *definiendum*. For instance, [8] introduced a method for answering definition questions that was assisted by a wrapper for the online Merriam Webster dictionary, which retrieved about

[1] http://experimental-quetal.dfki.de/

QAWeb 2008, April 22th, 2008, Beijing, China.
.

1.5 nuggets per question. These nuggets were used as query expansion terms for retrieving promising documents from the collection afterwards. Furthermore, they automatically constructed off-line a large relational database containing nuggets about every entity mentioned in the AQUAINT corpus. These nuggets were accordingly taken from every article within it, and therefore, answering definition questions consisted of a simple lookup for the *definiendum.* Since nuggets often seem odd and out of place without their context, they were expanded to surround one hundred (non-white-space) characters in order to enhance readability.

Unlike [9, 10], [8] took into account the filtering of redundant matched sentences by randomly removing one sentence from every pair that shared more than 60% of their terms. Unfortunately, this method discards relevant additional information placed along with the removed utterances, and does not account for sentences that are entirely overlapped with three or more phrases. It is also worth noting that this strategy answers definition questions in the TREC–2003 by aligning patterns at the word and the part-of-speech level.

Another example, is the strategy proposed by [3], which took advantage of external resources, like WordNet glossaries, online specific resources (e.g., Wikipedia), and web snippets for learning frequencies and correlation of words, especially with the *definiendum.* Candidate descriptive utterances were reranked according to their similarity to a centroid vector based upon these learnt frequencies. One of their findings was that definitional web-sites greatly improve the performance, leading to few unanswered questions: Wikipedia covered 34 out of the 50 TREC–2003 definition queries and biography.com 23 out of 30 questions regarding people, all together provided answers to 42 queries. They additionally found that web snippets, although they yielded relevant information about the *definiendum,* were not likely to supply descriptive utterances, bringing about only a marginal improvement.

Another strategy, proposed by[11], identifies windows of 250-characters that convey a definition. These windows are obtained from the top 50 documents retrieved by an IR engine and ranked by a SVM, which was trained using previously tagged windows according to the criteria of [10], and some automatically acquired phrasal attributes. The best configuration of their system obtained one acceptable definition within the top-five ranked windows for 116 out of 160 TREC–2000 questions and 116 out of 137 TREC–2001 questions. Later, [1] proposed an unsupervised version of this approach by extracting tagged windows from online encyclopedias.

Another method that takes advantage of web snippets was presented in [2]. This method uses a centroid vector that considers word dependencies learnt from the 350 most frequent stemmed co-occurring terms taken from the best 500 snippets retrieved by Google. These snippets were fetched by expanding the original query by means of a set of five highly co-occurring terms. These terms co-occur with the *definiendum* in sentences obtained by submitting the original query plus some task specific clues, e.g.,"*biography*". As a result, this query expansion technique improved the $\mathcal{F}(5)$ score of their system from 0.511 to 0.531. They concluded that the use of multiple search engines would help to fetch more sentences containing the *definiendum.*

The module of WebQA that answers definition questions was described firstly in [6]. Contrary to QAS in TREC, WebQA searches for definition sentences only on the web, in particular in web snippets. The advantage of descriptive phrases extracted from web snippets is that they provide an adequate unit of contextual information [6], being comparable in size with the enhanced nuggets obtained by [8]. For the purpose of markedly increasing the recall of definition sentences within web snippets, WebQA biases the search engine in favour of some lexico-syntactic structures that often convey definitions by means of a purpose-built query rewriting strategy. Then, WebQA clusters descriptive utterances according to *potential senses*, which are used to provide a partition of the most relevant and diverse utterances to the user. Results showed that WebQA is promising for answering definition questions in several languages directly from web snippets. In particular, WebQA found out descriptive information for all definition questions in the TREC 2001 and 2003 data sets. Specifically, WebQA finished with $\mathcal{F}(5)$ score of 0.53 for the TREC 2003 data-set, which is "competitive" with the best systems, which achieve a value between 0.5 and 0.56 [2, 8, 13, 14].

However, a key point for correctly interpreting these results is the completeness of the assessor's list. It is known that systems in TREC were able to find relevant nuggets, which were not included in this list (cf. [8] for details). In the case of web-based systems like WebQA, this vital fact is more likely to happen, because systems discover many additional nuggets seen as relevant by the user, but excluded from the assessor's list. This exclusion actually brings about a decrease in the $\mathcal{F}(5)$ score, because these extra nuggets enlarge the response without increasing precision. This kind of evaluation is, nonetheless, the unique current way to have an objective reference to the performance of several systems.

This study shows two search strategies that boost the recall of sentences that convey definitions, and consequently, they better the performance of the definition module of WebQA. These strategies: (a) take into consideration the prior knowledge provided by Google n-grams while rewriting the query, and (b) take up the suggestion of [2] by adding an extra search engine (Yahoo). Another thing minutely examined in this work, is the impact of the assessor's list on the evaluation of web-based definition QAS.

3. MINING THE WEB FOR DEFINITIONS

The definition component of WebQA receives the *definiendum* δ as input, assuming that it is previously identified by an external query analysis module or entered by the user. WebQA then proceeds as follows:

1. WebQA uses δ for rewriting Q according to a set Π of pre-defined surface patterns. These generated queries are then submitted to the search engine.

2. WebQA aligns patterns in Π with sentences extracted from the fetched snippets. Due to its complex internal structure [12], δ might match the *definiendum* δ' only partially within the retrieved descriptive utterances. Hence, WebQA recognises δ by means of relaxed pattern matching, based on the *Jaccard Measure.* The reason for using this relaxed matching strategy is that it provides WebQA with a higher degree of language independence compared to current definition QAS. In particular, we avoided the specification of additional word addition/ordering rules [12] or the integration of

more sophisticated linguistic processing such as chunking [8].

3. WebQA groups sentences by *potential senses*, which are discovered by observing the partitions generated by the closest neighbours of δ in the reliable semantic space supplied by Latent Semantic Analysis (LSA).

4. WebQA takes advantage of a variation of Multi-Document Maximal Marginal Relevance [7] for reducing redundancy and maximising diversity in selected utterances. This guarantees a fast summarisation framework which only makes use of a language–specific stop-list.

These four steps are described in the next sections in detail.

3.1 Obtaining descriptive sentences

In recent years, surface patterns for English have proven to be useful for distinguishing definition utterances in natural language texts [8, 9, 10, 11, 12]. These surface patterns provide syntactic structures that are properly aligned with sentences. These syntactic structures are, more precisely, based largely upon punctuation and words that often convey definitions (see table 1). Simply put, these syntactic structures make it possible to identify the *definiendum* $\delta^{'}$ and its definition nugget $\eta^{'}$ within utterances.

WebQA takes advantage of these syntactic structures not only for distinguishing definitions, but also for biasing the search engine in favour of web snippets that convey definitions. Currently, the ten search queries that help WebQA to substantially increase the recall of descriptive utterances within web snippets are as follows:

q_1=“δ”
q_2=“δ is a” $\vee$ “δ was a” $\vee$ “δ were a” $\vee$ “δ are a”
q_3=“δ is an” $\vee$ “δ was an” $\vee$ “δ were an” $\vee$ “δ are an”
q_4=“δ is the” $\vee$ “δ was the” $\vee$ “δ were the” $\vee$ “δ are the”
q_5=“δ has been a” $\vee$ “δ has been an” $\vee$ “δ has been the” $\vee$ “δ have been a” $\vee$ “δ have been an” $\vee$ “δ have been the”
q_6=“δ, a” $\vee$ “δ, an” $\vee$ “δ, the” $\vee$ “δ, or”
q_7=(“δ” $\vee$ “δ also” $\vee$ “δ is” $\vee$ “δ are”) $\wedge$ (called $\vee$ nicknamed $\vee$ “known as”)
q_8=“δ became” $\vee$ “δ become” $\vee$ “δ becomes”
q_9=“δ which” $\vee$ “δ that” $\vee$ “δ who”
q_{10}=“δ was born” $\vee$ “(δ)”

Once all snippets are fetched, WebQA removes all orthographic accents and splits them into sentences by means of intentional breaks and JavaRAP[2]. Patterns are then applied to discriminate descriptive utterances within retrieved snippets. Since δ does not exactly match $\delta^{'}$, WebQA takes advantage of the *Jaccard Measure* for distinguishing more reliable descriptive sentences. The *Jaccard Measure* (J) of two terms w_i, w_j, is the ratio between the number of different *uni-grams* that they share and the total number of different *uni-grams*:

$$J(w_i, w_j) = \frac{\mid w_i \cap w_j \mid}{\mid w_i \cup w_j \mid}$$

Consider, for example, the *definiendum* δ^*=“*George Bush*”, which might also be expressed as $\delta_1^{'*}$=“*George H. W. Bush*” or $\delta_2^{'*}$=“*Former US President Bush*”. The values for $J(\delta^*, \delta_1^{'*})$ and $J(\delta^*, \delta_2^{'*})$ are $\frac{1}{2}$ and $\frac{1}{5}$ respectively. WebQA filters reliable descriptive utterances by means of a pattern specific threshold (ψ_p). Of course, some sentences containing useful nuggets will be discarded, but these discarded nuggets can also be found in other retrieved phrases, e.g., “*Former US President Bush*” in “*George Bush was a former US President.*”. In short, WebQA implicitly trusts in the redundancy of the web for discovering several paraphrases.

3.2 Potential Senses Identification

There are many-to-many mappings between names and their concepts. On the one hand, the same name or word can refer to several meanings or entities. On the other hand, different names can indicate the same meaning or entity. For instance, δ=“*George Bush*” can refer to “*George H. W. Bush*” or “*George W. Bush*”.

WebQA disambiguates senses of δ by observing the correlation of its neighbours in the semantic space provided by LSA. This semantic space is constructed from the term-sentence matrix M, which considers δ as a *pseudo-sentence*. M is then weighted according to the traditional *tf-idf*. WebQA builds the dictionary of terms W from normalised elements in S. This normalisation consists of uppercasing, removal of html-tags, and the isolation of punctuation signs. WebQA then distinguishes all possible different *n-grams* in S together with their frequencies. The size of W is then reduced by removing *n-grams*, which are substrings of another equally frequent term.

WebQA makes use of $\hat{D}$, the greatest three eigenvalues of D, and the corresponding three vectors $\hat{U}$ and $\hat{V}$ for constructing the semantic space as $R = \hat{U}\hat{D}^2\hat{U}^{'}$. Then, WebQA uses the dot product as a measure of the semantic relatedness $R(w_i, w_j) = \hat{u}_i\hat{D}^2\hat{u}_j^{'}$ ($\hat{u}_i, \hat{u}_j \in \hat{U}$) of two terms $w_i, w_j \in W$. WebQA selects a set $\bar{W} \subseteq W$ of the forty highest closely related terms to δ. However, as a result of the relaxed pattern matching, WebQA must also account for all *n-grams* $\delta^+ \in W$ in δ, because some internal *n-grams* could be more likely to occur within descriptive utterances. WebQA considers, therefore, the forty highest pairs $\{w_i, R_{max}(\delta, w_i)\}$, where $R_{max}(\delta, w_i) = \max_{\delta+\in W} R(\delta^+, w_i)$. WebQA normalises terms in $\bar{W}$ according to:

$$\hat{R}(\delta, w_i) = \frac{R_{max}(\delta, w_i)}{\sum_{\forall w_j \in \bar{W}} R_{max}(\delta, w_j)}$$

Since words that indicate the same sense co-occur, WebQA identifies *potential senses* by finding a set $\bar{W}^\lambda \subseteq \bar{W}$ of words, for which their vectors form an orthonormal basis. In order to discriminate these orthonormal terms, WebQA builds a term-sentence matrix Φ, where a cell $\Phi_{is} = 1$, if the term $w_i \in \bar{W}$ occurs in the descriptive phrase $S_s \in S$, zero otherwise. The degree of correlation amongst words in $\bar{W}$ across S is then given by $\hat{\Phi} = \Phi\Phi^{'}$. Hence, the number of non-selected words $w_j \in \bar{W} - W^\lambda$ that co-occur with a term $w_i \in \bar{W}$ across S is given by:

$$\gamma(w_i) = \sum_{\forall w_j \in \bar{W} - \bar{W}^\lambda : \hat{\Phi}_{ij} > 0} 1$$

Then, WebQA adds the w_i to $\bar{W}^\lambda$ that:

$$\max_{w_i \in \bar{W}} \gamma(w_i) \qquad (1)$$

[2] http://www.comp.nus.edu.sg/∼qiul/NLPTools/JavaRAP.html

Table 1: Surface Patterns (Π).

π_1: $\delta^{'}$ [is|are|has been|have been|was|were] [a|the|an] $\eta^{'}$
e.g.,"***Noam Chomsky*** *is a linguist and social critic, professor at MIT, regular contributor to Z Magazine.*"

π_2: $[\delta^{'}|\eta^{'}]$, [a|an|the] $[\eta^{'}|\delta^{'}]$ [,|.]
e.g.,"***The new iPoD****, an MP3-Player,...* "

π_3: $\delta^{'}$ [become|became|becomes] $\eta^{'}$
e.g.,"***In 1996, Allen Iverson*** *became the smallest first-overall draft pick in the history of the NBA.*"

π_4: $\delta^{'}$ [,|] [which|that|who] $\eta^{'}$ [,|]
e.g.,"***Alberto Tomba****, who was the last Italian man to earn an Olympic skiing gold with victory in the 1992 giant slalom,..*"

π_5: $\delta^{'}$ [was born] $\eta^{'}$
e.g.,"***Niels Bohr*** *was born on 7th October 1885 in Copenhagen as the son of the physiologist Christian Bohr.*"

π_6: $[\delta^{'}|\eta^{'}]$, or $[\eta^{'}|\delta^{'}]$ [,|]
e.g.,"***Epilepsy****, or seizure disorder, refers to a group of disorders of the central nervous system...*"

π_7: $[\delta^{'}|\eta^{'}]$[|,][|also|is|are] [called|named|nicknamed|known as] $[\eta^{'}|\delta^{'}]$ [,|]
e.g.,"***Gordon Matthew Summer****, also known as 'Sting',...*"

π_8: $[\delta^{'}|\eta^{'}]$ $([\eta^{'}|\delta^{'}])$
e.g.,"***Euskadi Ta Askatasuna*** *(ETA)..*"

subject to:

$$\hat{\Phi}_{ij} = 0, \ \forall w_j \in \bar{W}^{\lambda} \tag{2}$$

$$\gamma(w_i) > 0 \tag{3}$$

In other words, a term w_i signals a new sense if it does not co-occur at the sentence level with any other already selected term $w_j \in \bar{W}^{\lambda}$, and it has the highest number of co-occurring non-selected terms $w_j \in \bar{W}$. Incidentally, WebQA breaks ties by randomly selecting a term. Words are added to $\bar{W}^{\lambda}$ until no other term w_i fulfils conditions (2) and (3). Next, sentences are divided into clusters C_{λ} according to terms in $\bar{W}^{\lambda}$. Sentences that do not contain any term in $\bar{W}^{\lambda}$ are collected in a special cluster C_0.

Finally, WebQA attempts to reassign each sentence S_s in C_0 by searching for the strongest correlation between its named entities (NEs) and the NEs of a cluster C_{λ}:

$$\max_{C_{\lambda}} \sum_{\forall e \in S_s} freq_{C_{\lambda}}(e) > 0, \quad \lambda \neq 0$$

where $freq_{C_{\lambda}}(e)$ is the frequency of NEs e in the cluster C_{λ}. The assumption here is that the same NEs tend to occur in the same sense.

3.3 Redundancy Removal

For each cluster C_{λ}, WebQA determines incrementally a set Θ_{λ} of its sentences S_{λ} to maximise their comparative relevant novelty:

$$\max_{S_s \in S_{\lambda} - \Theta_{\lambda}} coverage(S_s) + content(S_s)$$

subject to:

$$coverage(S_s) \geq \psi^{*} > 0 \tag{4}$$

$$W_{type}(S_s) = 0 \tag{5}$$

The comparative relevant novelty of a sentence S_s is given by the relative coverage and content of its nuggets respecting Θ_{λ}. Let $N(S_s)$ be the set of normalised nuggets associated with S_s and W_N then the set of terms of all normalised nuggets. $W_{N(S_s)}$ is the set of words in $N(S_s)$. Coverage is then defined as follows:

$$coverage(S_s) = \sum_{\forall w_i \in W_{N(S_s)} - W_{\Theta_{\lambda}}} P_i$$

where P_i is defined as the probability of finding a word $w_i \in W_N$, and is arbitrarily set to zero for all stop words. $W_{\Theta_{\lambda}}$ is the set of words occurring in preceding selected sentences Θ_{λ}.

Coverage is aimed at measuring how likely novel terms (not seen in Θ_{λ}) within $N(S_s)$ are to belong to a description. Thus, diverse sentences are preferred over sentences with many redundant words, which are consequently filtered according to an experimental threshold ψ^{*}. On the other hand, content discriminates the degree, in which $N(S_s)$ conveys definition aspects of δ based upon highly close semantic terms and entities, and is given by:

$$content(S_s) = \sum_{\forall w_i \in \bar{W}} \Phi_{is} \hat{R}(\delta, w_i) + \sum_{\forall e \in N(S_s) - E_{\lambda}} P_e^{\lambda}$$

The first sum measures the semantic bonding of terms in the respective nuggets, and the second sum the relevance of novel entities (E_{λ} is the set of entities in Θ_{λ}). Each novel entity e is weighed according to its probability P_e^{λ} of being in the normalised nuggets of C_{λ}. Incidentally, $W_{type}(S_s)$ is the amount of undesirable symbols in S_s such as pronouns, unclosed brackets or parenthesis, and URLs. Consequently, condition 5 bans sentences containing such symbols from Θ_{λ}. In summary, WebQA ranks sentences according to the order they are inserted into Θ_{λ}. This means that higher ranked sentences are more diverse, less redundant, and are more likely to contain entities along with terms that describe aspects of δ.

Note further that C_0 is processed last in order to initialise Θ_{λ} with all sentences selected from previous clusters, so that only sentences with novel pieces of information remain in C_0.

4. BOOSTING RECALL

The drawback to the query rewriting strategy presented in section 3.1 is that these search queries are statically built, causing that two promising lexico-syntactic clauses could be submitted in the same query, lessening the retrieval of descriptive phrases. A good illustrative example is δ="*George Bush*" and q_2. In this case, "*George Bush is a*" and "*George Bush was a*" are two clauses likely to yield definitions. Consequently, they should be separately submitted in order to avoid weakening the recall. Further, clauses such as "*George Bush were a*" and "*George Bush are a*" only bring about misleading sentences:

- *What if* ***George Bush were a*** *Black Man?*
- *If* ***George Bush were a*** *Democrat, many conservatives would be fighting him to the death.*
- *Born-again Christians like* ***George Bush are a*** *grave danger to the world, which you fail to even suggest in passing.*

Analogously, a set of unpromising lexico-syntactic patterns can be set in the same query and hence, bring about an unproductive retrieval, diminishing the number of descriptive utterances. Nevertheless, these patterns observe a local lexico-syntactic dependency with the *definiendum*, specifically, they are unlikely to contain additional words in between. This is an important fact, because off-line n-grams counts supplied by Google can be used to transform this static query construction into a more dynamic one. In our working example, an excerpt of Google 4-grams counts is as follows:

```
George Bush is a 20515
George Bush is an 3019
George Bush is the 10029
George Bush was a 2163
George Bush was an 240
George Bush was the 1810
George Bush are a 53
George Bush are an 44
George Bush are the 252
George Bush were a 103
George Bush were an -
George Bush were the 219
```

The first beneficial aspect of Google n-grams is that, in some cases, the grammatical number can be inferred. In particular, in the case of "*George Bush*", singular lexico-syntactic clues are most promising. However, it is not always possible to draw a clear distinction. A good example is "*fractals*":

fractals are a 176 (e.g. "Fractals are a powerful tool for modelling biological objects.")
fractals are an 86 (e.g. "Fractals are an exquisite interweaving of art and mathematics.")
fractals are the 215 (e.g. "Fractals are the place where math, science and art come...")
fractals is a 124 (e.g. "Fractals is a new branch of mathematics and art.)
fractals is the 148 (e.g. "Fractals is an innovative, class-leading. solution to the...)

Table 2: Dynamic queries (grammatical number known).

$q^{'}_7 = \emptyset$	$q^{'}_7 \neq \emptyset$
$q^{'}_1$:"δ R_1" $q^{'}_2$:"δ R_2" $q^{'}_3$:"δ R_3" $q^{'}_4$:"δ R_4" $q^{'}_5$:"δ R_5" $q^{'}_7$:"δR_6"	$q^{'}_1$:"δ R_1" $q^{'}_2$:"δ R_2" $q^{'}_3$:"δ R_3" $q^{'}_4$:"δ R_4" $q^{'}_5$:"δ R_5" $\vee$ "δ R_6"

Then, a strategy was designed, which selects a grammatical number whenever more than three keywords corresponding to one grammatical number exist, and zero to the another. The second favourable aspect is that the frequencies give hints about the hierarchy within the lexico-syntactic patterns. This method takes advantage of this hierarchy for configuring the ten queries. First, the search queries q_7 and q_{10} are merged into one query $q^{'}_7$. This query is composed of the following clauses:

"δ also called ", "δ also nicknamed", "δ also known",
"δ is called", "δ stands for", "δ is known",
"δ are called", "δ are nicknamed", "δ are known",
"δ was born", "δ was founded", "δ was founded",
"δ is nicknamed"

Accordingly, $q^{'}_7$ consists merely of the clauses that can be found in Google n-grams. If any clause cannot be found, $q^{'}_7$ is set to $\emptyset$. In any case, $q^{'}_{10}$ remains as $\emptyset$. It is worth pointing out that, the term "*stands for*" replaces the parentheses in q_{10}. Second, $q^{'}_5 = q_5$, $q^{'}_6 = q_6$ and $q^{'}_8 = q_8$ as well as $q^{'}_9 = q_9$. Additionally, the $q^{'}_1$ is set to $\emptyset$. Third, the clauses included in the queries q_2 and q_3, as well as q_4, are dynamically sorted across the available queries, as highlighted in table 2. In this table, R_1 and R_6 correspond to the highest and lowest frequent lexico-syntactic patterns according to Google frequency counts. In the case that the grammatical number cannot be distinguished, the queries are as follows:

$q^{'}_1$:"δ is a" $\vee$ "δ were an" $\vee$ "δ was the"
$q^{'}_2$:"δ was a" $\vee$ "δ are an"
$q^{'}_3$:"δ are a" $\vee$ "δ was an" $\vee$ "δ were the"
$q^{'}_4$:"δ were a" $\vee$ "δ is an"
$q^{'}_{10}$:"δ is the" $\vee$ "δ are the"

In the case $q^{'}_{10} = \emptyset$, the following three queries are reformulated:

$q^{'}_1$:"δ is a" $\vee$ "δ were an"
$q^{'}_3$:"δ are a" $\vee$ "δ was an"
$q^{'}_7$:"δ was the" $\vee$ "δ were the"

Every query is eventually surrounded with the feature "*inbody:*" in order to avoid matching a clause with the title of a web page.

5. EXPERIMENTS AND RESULTS

In [6], the definition module of WebQA was assessed by means of five question sets: (1) TREC 2001, (2) TREC 2003, (3) CLEF 2004, (4) CLEF 2005, and (5) CLEF 2006. The $\mathcal{F}(\beta)$-score [13] was accordingly computed as:

$$\mathcal{F}(\beta) = \frac{(\beta^2 + 1)(RP)}{\beta^2 P + R}$$

Where R and P stand for recall and precision, respectevely. Thresholds (ψ_p) for the specific surface patterns were all experimentally set to 0.25, apart from $\psi_1 = 0.33$ and $\psi_5 = 0.5$ (section 3.1). The threshold that controls redundancy ψ^*

Table 3: Results overview. (TQ = Total number of questions in the question-set)

Corpus		Baseline			WebQA			
	TQ	NAQ	NS	Accuracy	NAQ	NS	Accuracy	AS (%)
(1)	133	81	7.35 ± 6.89	0.87 ± 0.2	**133**	18.98 ± 5.17	0.94 ± 0.07	16 ± 20
(2)	50	38	7.7 ± 7.0	0.74 ± 0.2	**50**	14.14 ± 5.3	0.78 ± 0.16	5 ± 9
(3)	86	67	5.47 ± 4.24	0.83 ± 0.19	78	13.91 ± 6.25	0.85 ± 0.14	5 ± 9
(4)	185	160	11.08 ± 13.28	0.84 ± 0.2	173	13.86 ± 7.24	0.89 ± 0.15	4 ± 11
(5)	152	102	5.43 ± 5.85	0.85 ± 0.22	136	13.13 ± 6.56	0.86 ± 0.16	8 ± 14

was set to 0.01. As WebQA, the implemented Baseline retrieves 300 hundred snippets by submitting q_1 and splits them into sentences by means of intentional breaks and JavaRAP afterwards. Baseline accounts solely for a stricter matching of δ by setting all pattern thresholds $\psi_p = 1$. A random sentence from a pair that shares more than 60% of their terms is discarded, cf. [8], as well as sentences that are a substring of another sentence. No clustering of sentences by potential senses is done.

Coverage.

Table 3 shows the coverage of Baseline and WebQA. NAQ stands for the number of questions, for which its response contained at least one nugget (manually checked). WebQA discovered nuggets for all questions in (2), contrary to [3], who found nuggets for solely 42 questions by using external dictionaries and web snippets. In addition, WebQA discovered nuggets within snippets for the 133 questions in (1), in contrast to [11], who found a top five ranked snippet that convey a definition solely for 116 questions within top 50 downloaded full documents. Additionally, WebQA extracts short sentences[3] (125.7 ± 44.21 considering white spaces; Baseline: 118.168 ±50.2), whereby [11, 1] handled fixed windows of 250 characters. On the other hand, sentences found by WebQA are 109.74 ± 42.15 characters long without considering white spaces, which is comparative longer than the 100 characters nuggets of [8], who fetched 1.5 nuggets per definition by means of specialised wrappers. A final remark regarding lengths, the length of descriptive sentences discovered by Baseline was 118.168 ±50.2 considering white spaces, and 97.81 ±41.8 without white spaces. Due to the acceptable length of descriptive sentences and the fact that a lot nuggets seems odd without their context, WebQA prefers to output sentences instead of only nuggets.

Overall, WebQA covered 94% of the questions, whereas Baseline 74%. This difference is mainly due to the query rewriting step and the more flexible matching of δ. For all questions, in which WebQA and Baseline discovered at least one nugget, the accuracy and the average number of sentences (NS), containing also at least one nugget, was computed. WebQA doubles the number of sentences and achieves a slightly better accuracy. In order to compare the diversity of both responses, the ratio of the number of words in W_{Θ_λ} excluding stop words to the number of sentences in Θ_λ was computed: Baseline=6.47 ± 1.75 and WebQA=8.30 ± 1.44. In table 3, AS corresponds to the percentage of sentences within NS, for which the relaxed matching shifted δ to another concept. Some shifts brought about interesting descriptive sentences. A good examples is: "*neuropathy*" was shifted to "*peripheral neuropathy*" and "*diabetic neuropathy*", conversely, some shifts caused unrelated sentences: "*G7*" to "*Powershot G7*".

Table 4: TREC 2003 results.

	Recall (R)	Precision (P)	Average length
Baseline	0.35 ± 0.34	0.30 ± 0.26	583
WebQA	0.61 ± 0.33	0.18 ± 0.13	1878

TREC 2003.

In order to compare our methods with a gold standard, we used the assessors' list provided by the TREC 2003 data. Following the approach in [13], table 4 displays our current achievement. Given the higher recall 0.61 ± 0.33 obtained by WebQA, it can be concluded that the additional sentences that it selects contain more nuggets seen as vital on the assessor's list. The $\mathcal{F}(\beta)$ was accordingly computed for each response:

Table 5: TREC 2003 $\mathcal{F}(\beta)$ scores.

β	1	2	3	4	5
Def-WQA	0.26	0.37	0.45	0.50	0.53
Baseline	0.26	0.30	0.32	0.32	0.34

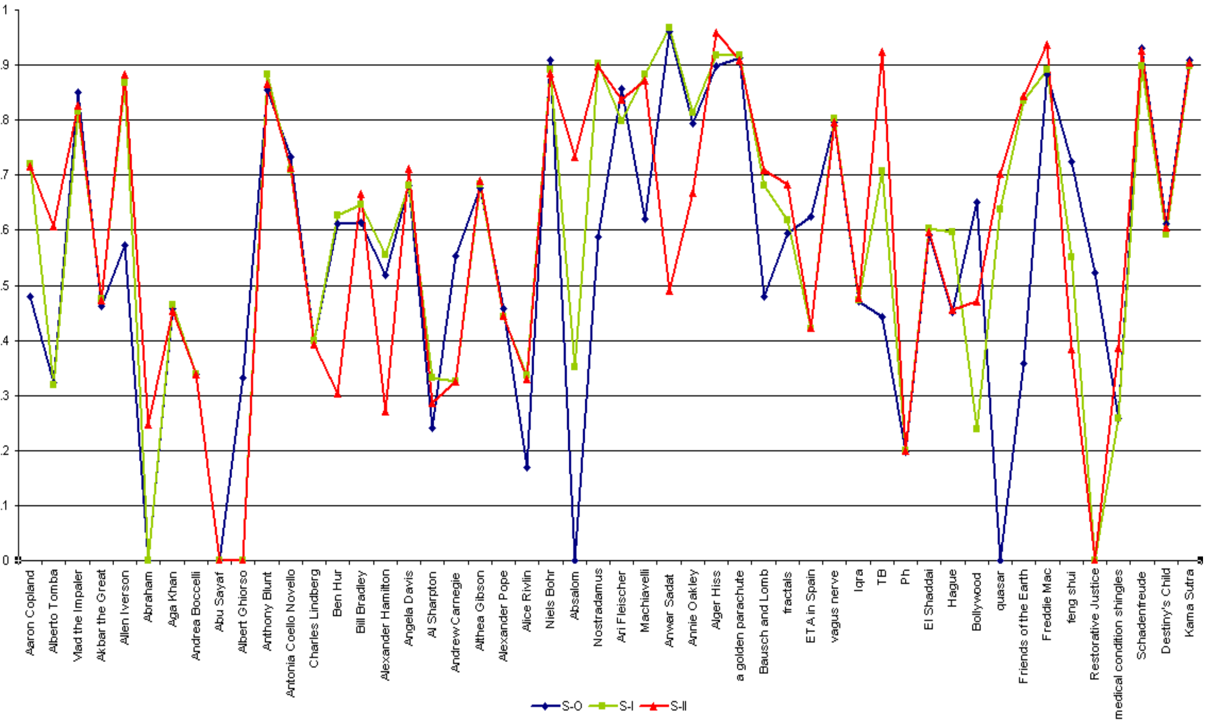
WebQA was able to distinguish different potential senses for some δs, e.g., for "*atom*", the particle–sense and the format–sense. On the other hand, some senses were split into two separate senses, e.g., "*Akbar the Great*", where "*emperor*" and "*empire*" indicated different senses. This misinterpretation is due to the independent co-occurrence of "*emperor*" and "*empire*" with δ, and the fact that they are not likely to share words. In order to improve this, some external sources of knowledge are necessary. Like [2], we noticed that this is an extreme hard problem, because some δs can be extremely ambiguous like "*Jim Clark*", which refers to more than ten different real-world entities. In this case, WebQA can differentiate the photographer, the pilot, the Netscape creator, but many executive named "*Jim Clark*" are grouped in the same cluster.

TREC 2003: Boosting Recall.

The dynamic (S-I) and the static (S-O) query rewriting strategies were assessed by means of the defintion question set supplied by TREC 2003. Following the suggestion of [2], S-I was additionally tested together with the use of an extra search engine (S-II). Figure 1 compares the $\mathcal{F}(5)$ score per

[3] Along this section, ± stands for standard deviation, and CLEF data-sets consider all English translations from all languages.

Figure 1: Comparison between $\mathcal{F}(5)$ scores obtained by each strategy for each *definiendum* in the TREC 2003 question-set.



question for the three strategies. WebQA with the static query rewriting finished with an average $\mathcal{F}(5)$ score of 0.5472, while the dynamic query rewriting improved the average value to 0.5792, and this rewriting along with an additional search engine, improved to 0.5842. Here, it is worth remarking that S-I obtained an improvement without increasing the number of submitted queries, whereas the marginal increase achieved by S-II with respect to S-I, is at the expense of sending ten extra queries to the additional search engine. Overall, the $\mathcal{F}(5)$ values, achieved by WebQA with our rewriting strategies incorporated, are "competitive" with the best definition QAS. These systems obtain a value between 0.5 and 0.56 [2, 8, 13, 14].

Error Analysis: Future Challenges.

S-O and S-I scored zero for four different *definiendums*, despite the "okay" nuggets found by both systems. In fact, if a system does not discover any nugget assessed as "vital", it finishes with a $\mathcal{F}(5)$ value equal to zero. For instance, S-II scored zero for three questions; in particular, for the following output concerning "*Albert Ghiorso*":

- *said* ***Albert Ghiorso, a*** *veteran Berkeley researcher, who holds the Guinness world record.*
- ***Albert Ghiorso is a*** <u>*nuclear scientist*</u> *at Lawrence Berkeley National Laboratory in Berkeley, Calif.*
- *That's what Berkeley Lab's* ***Albert Ghiorso, a*** *man who has participated in the discovery of more atomic elements than any living person, told the students and teachers who packed.*
- ***Albert Ghiorso is an*** *American* <u>*nuclear scientist*</u> *who helped discover several elements on the periodic table.*

The "okay" nugget is underlined that matches the assessors' list provided by TREC 2003:

vital designed and built cyclotron accelator
okay nuclear physicists/experimentalist
vital co-creator of 12 artificial elements
vital co-discovered element 106

Like [8] also noticed, "okay" nuggets, like *nuclear physicists/experimentalist* can be easily interpreted as "vital". For example, if one considers abstracts supplied by Wikipedia as a third-party judgement, at the time of writing, one finds:

- ***Albert Ghiorso*** *(b. 15 July 1915) is an American* <u>*nuclear scientist*</u> *who helped discover numerous chemical elements on the periodic table.*

Further, some relevant nuggets, including *veteran Berkeley researcher*, are unconsidered, enlarging the response, and thus decreasing the $\mathcal{F}(5)$ score. We hypothesise that a nugget can be seen as "vital" or "okay" according to how often its type (birthplace, birthdate, occupation, outstanding achievement) occurs across abstracts of online encyclopedias, such as Encarta or Wikpedia. We deem that this sort of type-oriented evaluation would be more appropriate to web-based definition QAS. Only in one *definiendum* were the three strategies unable to discover any nugget in the assessor' list: "*Abu Sayaf*". The reason is uncovered when the following frequencies on Google n-grams are checked:

```
Abu Sayyaf 96204
Abu Sayyafs 89
Abu Sayaf 1156
Abu Sayaff 3205
```

In this case, the spelling of the *definiendum* in the query is unlikely to occur in the web, causing an $\mathcal{F}(5)$ equals to zero. Conversely, when WebQA processes "*Abu Sayyaf*", the scores obtained by each method are: 0.844 (S-O), 0.8794 (S-I) and 0.8959 (S-II). Accordingly, the new average $\mathcal{F}(5)$ values are: 0.564 (S-O), 0.59679 (S-I) and 0.602 (S-II).

Another complicated problem is that the list of the assessor is aimed predominantly at one possible sense of the *definiendum*. Hence, discovered descriptive utterances concerning additional senses, similar to the uncosidered nuggets, bring about a decrease in the $\mathcal{F}(5)$ value. To illustrate this, a descriptive sentence found by S-II regarding "*Nostradamus*":

- ***Nostradamus is a*** *neural network-based, short-term demand and price forecasting system, utilized by electric and gas utilities, system operators and power pools...*

Indeed, it is highly frequent to find ambiguous terms. For example, Wikipedia contains more than 19000 different disambiguation pages. In this case, the list of the assessor only accounts for the reference to the French astrologer/prophet. When sentences concerning other senses are manually removed, the $\mathcal{F}(5)$ values for this concept increase as follows: from 0.5871 to 0.5936 (S-O), from 0.9028 to 0.9182 (S-I) and from 0.8977 to 0.9167 (S-II). However, at the time of writing, Wikipedia does not disambiguate "*Nostradamus*", but it provides disambiguation pages for eighteen out of the fifty TREC 2003 *definiendums* including "*Ben Hur*", and "*Kamasutra*". Obviously, a more noticeable difference in $\mathcal{F}(5)$ score is due to *definiendums* with more senses such as "*Absalom*". Incidentally, it is also worth remarking that Wikipedia did not supply definitional information for two *definiendums*: "*Alexander Hamilton*" and "*medical condition shiggles*".

Another difficulty that QAS encounter when they extract definition phrases from the web, is that opinions are also given like definitions. A good example is given by the *definiendum* "*Charles Lindberg*":

- ***Charles Lindberg was a*** *true American hero.*

This sentence does not syntactically differ from the definition "*Charles Lindberg was a famous American pilot.*" We envisage that a large-scale redundancy and the use of opinion mining techniques would help to discriminate opinions from facts.

6. CONCLUSIONS AND FUTURE WORK

Our ongoing research is aimed at incorporating more linguistic information into the query rewriting strategy. Specifically, promising verb phrases can be interpreted as definition lexico-syntactic patterns, and therefore, appended to the *definiendum*. These verb phrases can be determined by means of retrieved descriptive sentences, a chunker, and the corresponding recalls can be estimated by inspecting the frequency of these new clauses on Google n-grams. This sort of strategy would help to fetch more and diverse descriptive information about the *definiendum*.

This study compares three query rewriting strategies that are aimed at boosting the recall of descriptive sentences in web snippets and consequently, at improving the performance of definition QAS. One interesting finding is that Google n-grams can be used particularly for optimising the retrieval of defintions in web snippets, and accordingly, they can also assist QAS in fetching more promising full documents.

This paper additionally discusses the major challenges posed by web-based definition QAS, and it sketches accordingly some directions that could help to face these challenges. In particular, frequencies of types of nuggets occurring across abstracts in Wikipedia would assist in carrying out a more objective evaluation of web-based definition QAS.

7. ACKNOWLEDGMENTS

This work was partially supported by a research grant from the German Federal Ministry of Education, Science, Research and Technology (BMBF) to the DFKI project `HyLaP` (FKZ: 01 IW F02) and the EC-funded project QALL-ME. Part of this work is copyrighted by Springer-Verlag.

8. REFERENCES

[1] I. Androutsopoulos and D. Galanis. A practically unsupervised learning method to identify single-snippet answers to definition questions on the web. In *HLT/EMNLP*, pages 323–330, 2004.

[2] Y. Chen, M. Zhon, and S. Wang. Reranking answers for definitional qa using language modeling. In *Coling/ACL-2006*, pages 1081–1088, 2006.

[3] T. S. C. H. Cui, M. Kan, and J. Xiao. A comparative study on sentence retrieval for definitional question answering. In *SIGIR Workshop on Information Retrieval for Question Answering (IR4QA)*, 2004.

[4] A. Figueroa and G. Neumann. Language independent answer prediction from the web. In *FinTAL 5th International Conference on Natural Language Processing*, 2006.

[5] A. Figueroa and G. Neumann. Mining web snippets to answer list questions. In *AI07: the 2nd International Workshop on Integrating AI and Data Mining*, 2007.

[6] A. Figueroa and G. Neumann. A multilingual framework for searching definitions on web snippets. In *KI 2007: Advances in Artificial Intelligence*, pages 144–159, 2007.

[7] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. Multi-document summarization by sentence extraction. In *NAACL-ANLP 2000 Workshop on Automatic summarization*, pages 40–48, 2000.

[8] W. Hildebrandt, B. Katz, and J. Lin. Answering definition questions using multiple knowledge sources. In *HLT-NAACL 2004*, pages 49–56, 2004.

[9] H. Joho and M. Sanderson. Retrieving descriptive phrases from large amounts of free text. In *9th ACM conference on Information and Knowledge Management*, pages 180–186, 2000.

[10] H. Joho and M. Sanderson. Large scale testing of a descriptive phrase finder. In *First Human Language Technology Conference*, pages 219–221, 2001.

[11] S. Miliaraki and I. Androutsopoulos. Learning to identify single-snippet answers to definition questions. In *First Human Language Technology Conference*, pages 1360–1366, 2004.

[12] M. M. Soubbotin. Patterns of potential answer expressions as clues to the right answers. In *TREC-10 Conferenc*, pages 1360–1366, 2002.

[13] E. M. Voorhees. Evaluating answers to definition questions. In *First Human Language Technology Conference*, pages 109–111, 2003.

[14] J. Xu, A. Licuanan, and R. Weischedel. Trec2003 qa at bbn: Answering definitional questions. In *Twelfth Text REtrieval Conference*, 2003.