Mining Wikipedia for Discovering Multilingual Definitions on the Web

Alejandro Figueroa

German Centre for Artificial Intelligence - DFKI
Stuhlsatzenhausweg 3, D - 66123, Saarbrücken, Germany
figueroa@dfki.de

***Abstract*— Ml-DfWebQA is a multilingual definition question answering system (QAS) that extracts answers to definition queries from the short descriptions of web-sites returned by search engines, called web snippets. These answers are discriminated on the ground of lexico-syntactic regularities mined from multilingual resources supplied by Wikipedia. Results support that these regularities serve to significantly strengthen the answering process. In addition, Ml-DfWebQA increases the robustness of multilingual definition QASs by making use of aliases found in Wikipedia.**

I. Introduction

Definition questions, such as "*What is SKG2008?*" and "*Who is Nigel Farage?*", have become especially interesting in recent years, because about 27% of the questions in real user logs are requests for definitions, as well as about 25% of the queries submitted to a search engine [1]. Simply put, definition questions are queries aimed at learning more about a topic or concept: "find out about" and "tell me about".

By and large, QASs ground their answering strategy on the extraction of definition information from specific resources (e.g., Wikipedia), and/or significant word correlations with the target concept. These correlation statistics are taken normally from sentences containing the target concept across a corpus. Definition QASs are consequently dependent upon the coverage of specific resources and/or upon finding strong correlation patterns, and thus they tend to miss descriptive information low in frequency.

This work deals with this problem by inferring lexico-syntactic regularities from definitions corresponding to similar concepts. In other words, Ml-DfWebQA naturally assumes that definitions of two similar concepts, such as politicians, would share some similarities, and hence these similarities can assist in recognising descriptive information low in frequency. Ml-DfWebQA learns these regularities from Wikipedia. The present work additionally introduces an approach for tackling common concept aliases/mispellings head-on.

Ml-DfWebQA is aimed at identifying answers within web snippets in several languages, namely English and Spanish. Aiming at Spanish is particularly interesting, because the coverage yielded by the web and specific resources substantially differs with English, and it is, for this reason, more difficult to recognise definitions in one language than the other. Incidentally, the motivation behind the use of web snippets as an answer source is two-fold: (a) to avoid the costly retrieval and processing of full web documents, and (b) to the user, web snippets are the first view of the response, thus highlighting answers would make them more informative. Another strong incentive is that the absence of answers across web snippets can force a request for extra feedback at the user.

II. Related Work

QASs 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. In order to discover correct answers in this corpus, QAS extracts nuggets from several external resources of definition information. QAS then identify descriptive phrases by projecting the obtained nuggets into the corpus. In the jargon of definition questions, a nugget ("*Semantic Context Unit*" or SCU) is a piece of relevant or factual information about the topic of the question (aka. *definiendum* or target).

For instance, [2] took advantage of external resources like WordNet glossaries, online specific resources (e.g., Wikipedia) and web snippets for learning frequencies of words that correlates with the *definiendum*, which were used for forming a centroid vector afterwards. Sentences were thereafter ranked according to the cosine distance to this vector. One of their findings was that definitional web-sites greatly enhance 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 providing answers to 42 queries. They additionally found that web snippets, though they yielded relevant information about the *definiendum*, were not likely to supply descriptive utterances, bringing about only a marginal improvement. On the one hand, these specific resources provide accurate and succinct information about the *definiendum*, on the other hand, if the system only makes use of these resources many questions will not be covered, or limited covered. This limitation on coverage reaffirms the need to account for the web as a source of descriptive information.

[3] introduced another strategy that takes advantage of web snippets. 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*"). We reasonably deem that these task specific clues are aimed at biasing the search in favour of snippets taken from several online specific resources. The 350 words are then used for building an ordered centroid vector by retaining their original order within the sentences. Then, these ordered centroid vectors are used for training language models, which are later utilised for ranking candidate answers.

At this point, it is worth bearing in mind that there are two aspects that make [3] and [2] less attractive: (a) both approaches rely strongly on finding entries for the *definiendum* in several specific resources, making it possible to count reliable frequencies of word correlations with the *definiendum*, and (b) it is hard to detect SCUs expressed with words lowly correlated with the *definiendum*, the inevitable consequence is thus a less diverse output and detrimental to coverage.

[2] and [3] are aimed at discovering answers in the AQUAINT corpus, while Google[1] offers a feature for searching definitions on the web. Every time a user enters "define:*definiendum*", the search engine returns a set of glossaries containing definitions of the term. To the best of our knowledge, it is hitherto unknown how Google gathers these glossaries: Which strategies are involved? What is manual or automatic? However, [4] observed that these glossaries seem to have some common properties: the pages are entitled with task specific clues including "*glossary*" and "*dictionary*", the terms in the page are alphabetically sorted and presented with the same style, for instance italics and bold print. Under this observation, this method yields wider coverage, but succinct definitions taken from different glossaries are very likely to convey redundant information, while at the same time, new concepts are rarely found in glossaries, but in web-sites such as blogs or forums.

[5], [6] proposed a strategy designed to largely overcome this coverage problem. They substantially boosted the recall of descriptive sentences within web snippets by rewriting the query, in such a way that there is an increased probability of matching definition patterns. This query rewriting strategy is based on some lexico-syntactic constructions that are commonly used and very likely to convey and recognise definitions[6], [7], [8], [9]. This method discovered SCUs for all 50 questions in TREC 2003, and for 570 out of 606 CLEF questions, proving that web snippets are a promising source of definition phrases, and hence challenging the finding of [2]. This approach, however, still depends entirely on word correlation counts when it ranks and selects definitions, and therefore it is also hard for this method to distinguish descriptive nuggets expressed with words low in frequency. The attractive facet of this strategy is, nevertheless, that it is aimed at discovering definitions on the entire web, determining their likelihood of being answers from their contextual evidence, this means it does not project answer candidates into a target corpus. In this approach, however, every time the user inputs an alias of the *definiendum* or a common misspelling, the performance of this strategy is diminished.

For additional work on definition discovery in English, the reader can also, for example, see [4], [7], [8], [9]. While TREC has focussed its attention on English, the Cross-Language Evaluation Forum (CLEF) has been giving attention to European languages including Spanish. In the Spanish track, QASs are challenged to find answers across the EFE corpus, which comprises about 450,000 documents. In particular, the best system, regarding definition questions, makes use of lexico-syntactic constructs [10] and entries in Wikipedia [11], continuing the trend of QASs in TREC. Contrary to CLEF systems, [6] extracted answers to definition questions in Spanish from the web by extending their system to handle this class of question. As a result, they showed that it is plausible to extract answers to definition queries from web snippets for several languages.

Our Contribution: `Ml-DfWebQA` improves the efficiency and robustness of definition QASs by means of lexico-syntactic regularities learnt from Wikipedia. These learnt regularities assist in recognising SCUs low in frequency. In addition, `Ml-DfWebQA` takes advantage of *definiendum* aliases, provided by Wikipedia, for improving robustness.

III. Mining Multilingual Wikipedia Resources

A. Aliases Repository

Wikipedia[2] classifies its pages according to their content into the following groups: redirection, disambiguation, definition, list, or categories. In this classification scheme, redirection pages contain no definition content, but they are used for linking an input string with its respective definition page. In this work, these input strings are interpreted as aliases of the respective main concept. To illustrate this, the redirection page of "*Nicolas Sarcozy*" connects this alias to the definition page of "*Nicolas Sarkozy*". These mappings are used for building an **off-line** repository of aliases:

<Nicolas Sarkozy, President Sarkozy, redirection, en>
<Nicolas Sarkozy, Nicolas Sarcozy, redirection, en>
<Nicolas Sarkozy, Nicolas Sarközy, redirection, en>
<Nicolas Sarkozy, Nicolás Sarkozy, redirection, es>

This repository is additionally enriched with aliases conveyed in first definition phrases. Consider the following example corresponding to the abbreviation "*MSN*":

*"**MSN**" (an abbreviation for "**Microsoft Network**") is a collection of Internet services provided by Microsoft.*
*"**MSN**" (abreviación de "**Microsoft Network**") es una colección de servicios de internet proporcionado por Microsoft.*

Sentences containing aliases are discriminated **off-line** on the grounds of pre-defined lexico-syntactic clues. These clues

[1] http://www.google.com/help/features.html#definitions

[2] In the scope of this work, we use the snapshot supplied by Wikipedia in January 2008.

have determined by inspecting n-grams high in frequency, occurring in these phrases, that indicate aliases. Good examples are "*also known as*" and "*an acronym for*" in English, while in Spanish good examples are "*más conocido como*" and "*abreviación de*". In our previous example, the next mappings are obtained:

<MSN, Microsoft Network, first line, en>
<MSN, Microsoft Network, first line, es>

Then, finding out the aliases of a particular concept consists of looking for the right entry in this repository.

B. Definition Templates Repository

To begin with, a corpus C is extracted consisting of the phrases, in the abstracts of Wikipedia, that match definition patterns. These patterns are applied at the sentence and surface level, and comprise well-known lexico-syntactic constructs [6], [7], [8], [9]. Secondly, entities in C are discriminated on the ground of word sequences that begin with a capital letter, and a name entity recognizer [3]. Afterwards, identified name entities are replaced with a placeholder (#).

Thirdly, bigrams to decagrams are extracted from the definition part of these modified first sentences. We call these resulting n-grams templates, and only templates that start at any of the first four words are considered. Lastly, an histogram of templates is built (see tables I and II), and templates with a frequency lower than six are discarded. It is worth highlighting here that the initial reduction in variation, due to the replacement of name entities by a placeholder, helps to obtain more reliable template counts.

The basic idea behind this off-line repository is that these templates are not only highly likely to indicate definitions, but also to start these descriptions. Take, for instance, the following two definitions taken from web snippets:

Daniel Hannan **is a** British **politician** who is currently..
Angela Dorothea Merkel (born July 17, 1954 in Hamburg) **is a** German **politician** and the conservative opposition's..

Here, "*is a British politician*" and "*is a German politician*" match the relatively high in frequency template "*is a # politician*" (see table I), and it consequently helps to distinguish these descriptive phrases without needing to check whether or not an entry in a specific resource exists.

C. Definition Tuples Repository

[12] computed word association norms directly from unstructured natural language text. They proposed a measure, named *association ratio*, grounded on the idea of mutual information. The *association ratio* (I_2) between two words w_1 and w_2 is defined as:

$$I_2(w_1, w_2) = log_2 \frac{P(w_1, w_2)}{P(w_1)P(w_2)}$$

[3]For this purpose, we use Stanford NER publicly available at: http://nlp.stanford.edu/software/CRF-NER.shtml

TABLE I
SAMPLE INTERESTING TEMPLATES IN ENGLISH.

t	$len(t)$	$freq(t)$
is a species of	4	34878
member of the #	4	23351
a # politician	3	13422
a municipality in the district of #	7	8922
is a # politician	4	4776
is a # politician and the	6	162
is a # politician who is currently	7	18

TABLE II
SAMPLE INTERESTING TEMPLATES IN SPANISH.

t	$len(t)$	$freq(t)$
y comuna francesa en la region de #	8	3330
es una comuna y poblacion de # en la region	10	3310
es un municipio de la	5	2976
es un politico	3	1471
un club de futbol	4	1452

This ratio compares the probability of observing w_2 followed by w_1 within a fixed window of k words with the probabilities of observing w_1 and w_2 independently. This ratio differs from mutual information in the encoded linear precedence, and captures some lexico-syntactic regularities in the target corpus [12].

For the remainder of this paper, this ratio is computed on the definition parts of the phrases in C by making allowances for a window size of ten, and the probabilities are estimated as described in [12]. Since this ratio becomes unstable when counts are very small, like [12], word pairs with a frequency lower than six were discarded. In addition, pairs consisting solely of stop-words[4] were also filtered out.

Under the underpinning assumption that relevant pairs will exhibit a joint probability larger than the product of the probability of finding them by chance. In our work, this word association ratio is extended to triples as follows:

$$I_3(w_1, w_2, w_3) = log_2 \frac{P(w_1, w_2, w_3)}{P(w_1)P(w_2)P(w_3)}$$

Like [12], we noticed the larger the ratio is, the more credible results it computes. Conversely, the values become less interesting while the ratio approaches to zero. Negative ratios are rare, but possible, and [12] suggest that it indicates a complementary relationship. Simply put, this ratio supplies a efficient way to identify some semantic and lexico-syntactic relations.

Table III emphasises some interesting tuples, in this repository of aliases, concerning the word w_*="*politician*" ($freq(w_*) = 32306$), some of these tuples can help to identify the working descriptive phrases shown in section III-B.

[4]We use the 319 highly frequent close class forms contained in http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words.

TABLE III

SOME ASSOCIATIONS WITH "POLITICIAN".

$\vec{w} =< w_1, w_2 >$	$I_2(\vec{w})$	$\vec{w} =< w_1, w_2, w_3 >$	$I_3(\vec{w})$
$< w_*$, diplomat>	7.06	$< $ a, w_*, currently>	7.41
$< w_*$, currently>	4.33	$< w_*$, who, currently>	7.14
$< w_*$, opposition>	4.15	$< $ a, w_*, conservative>	2.93
$< w_*$, conservative>	3.44	$< $ a, w_*, opposition>	2.71
$< w_*$, coach>	-0.30	$< w_*$, the, junior>	-5.08

Table IV stresses some associations with w_*="*escritor*" discovered in Spanish. This table highlights a beneficial aspect of this *association ratio*: it can be computed for many languages.

TABLE IV

SOME ASSOCIATIONS WITH "ESCRITOR".

$\vec{w} =< w_1, w_2 >$	$I_2(\vec{w})$	$\vec{w} =< w_1, w_2, w_3 >$	$I_3(\vec{w})$
$< w_*$, cuentos>	8.77	$< w_*$, columnista, de >	26.19
$< w_*$, critico>	8.47	$< w_*$, cuentos, relatos>	22.11
$< w_*$, autor>	6.71	$< w_*$, ciencia, ficcion>	18.84
$< w_*$, famoso>	5.33	$< w_*$, mas, importantes>	6.76
$< w_*$, grupo>	0.12	$< w_*$, de, religion>	-2.83

IV. ANSWERING DEFINITION QUESTIONS

Ml−DfWebQA answers definition questions in two sequential steps: descriptive sentences retrieval and ranking. The former includes a query rewriting strategy that boosts the retrieval of definition phrases (section IV-A). The latter involves the selection of a set of relevant and diverse descriptive sentences (section IV-B).

A. Searching for definitions

Ml−DfWebQA searches for definition phrases as follows:

1) Rewrite the query using the copular lexico-syntactic structures as shown in [5], [6]. If more than fifty definition phrases are identified, then proceed to step three, otherwise step two.
2) Select a *definiendum* alias (see section IV-A.1), and obtain its respective descriptive phrases afterwards.
3) In the case of English, obtain additional descriptive sentences by means of the queries sketched in section IV-A.2. In the case of Spanish, Ml−DfWebQA retrieves extra descriptive sentences by sending the remaining queries in [6].

It is worth remarking here that definition phrases are distinguished at the surface level as described in [6].

1) Selecting an alternative definiendum: Web definition QAS are occasionally unable to find descriptive information, because the spelling of the *definiendum* entered by the user is unlikely to occur in the web. First, Ml−DfWebQA tackles this problem by examining candidate aliases within the aliases repository. Ml−DfWebQA selects candidates discovered in the first definition lines, and in the case that nothing is found, within aliases extracted from redirections, thereby ensuring that most reliable aliases are considered firstly. Due to the query length restrictions imposed by search engines, Ml−DfWebQA chooses aliases candidates written with two or three words. The more promising candidates aliases are then selected as follows:

1) If the inputted *definiendum* is formed of three words, Ml−DfWebQA picks aliases that consider the removal of one term. For instance, "*Angela Merkel*" would be considered if the input is "*Angela Dorothea Merkel*".
2) Aliases containing the same number of words, such as "*Nicolas Sarkozy*"⇔"*Nicolas Sarcozy*", are considered.
3) If the alias resolves or corresponds to an acronym.
4) Only aliases that contain letters and numbers, hyphen, spaces and/or ampersand, are taken into account.

For the purpose of selecting the right replacement, Ml−DfWebQA sends the search engine five search queries per alias candidate. These five purpose-built queries were proposed by [5] (English) and [6] (Spanish), and each submission is aimed at a maximum of 30 snippets. Ml−DfWebQA reasonably assumes that clearer evidence (more descriptive phrases) will come into light in the case of the most promising alias.

2) Template Query Generation: Ml−DfWebQA obtains search clauses from Google 5-grams[5] by searching for n-grams that start with the *definiendum* (δ). Then, Ml−DfWebQA[6] sees a 5-gram as a search clause, if its respective template structure matches an element in the repository of templates (section III-B). Here, contrary to the construction of the repositories, Ml−DfWebQA does not account for a Name Entity Recogniser. Search clauses are hence ranked according to their Google 5-grams frequency. Some examples are the search clauses with respect to "*Angela Merkel*":

Angela Merkel , the conservative 112
Angela Merkel , the leader 319
Angela Merkel , the opposition 53
Angela Merkel , who makes 57
Angela Merkel , who took 48

By default, based on the spirit of [5] and [6], Ml−DfWebQA boosts the retrieval of snippets containing descriptive phrases by taking advantage of the following five queries:

q_1: "δ, a" ∨ "δ, an" ∨ "δ, the" ∨ "δ or"
q_2: "δ which" ∨ "δ who" ∨ "δ that"
q_3: "δ becomes" ∨ "δ become" ∨ "δ became"
q_4: "δ has been a" ∨ "δ has been the" ∨ "δ has been a" ∨ "δ has been the"
q_5: "δ was founded" ∨ "δ was born" ∨ "δ was grounded" ∨ "δ stands for"

However, Ml−DfWebQA modifies q_2 and q_3 as well as q_4 to increase the probability of fetching definition phrases by means of the search clauses. For example:

q_2: "Angela Merkel, who makes" ∨ "Angela Merkel, who took"

[5]http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html

[6]The reason to only apply this query expansion strategy to English is that Google n-grams are only available for English.

The idea behind this modification is trying to focus directly on more specific clues that are very likely to express definitions. Since simultaneously matching all clauses in q_5 is unlikely, if there is less than four specific clues, the query is formed by adding the remaining high in frequency search clauses. Some examples of q_5 are:

q_5: "Angela Merkel, the conservative" $\vee$ "Angela Merkel, the leader" $\vee$ "Angela Merkel, the opposition"

q_5: "Alexander Hamilton was born in" $\vee$ "Alexander Hamilton was born on" $\vee$ "Alexander Hamilton , a founding" $\vee$ "Alexander Hamilton who served at"

Queries are constrained by the length imposed by the search engine, and aimed at a maximum of 30 snippets each.

B. Ranking definitions

`Ml-DfWebQA` ranks the descriptive phrases D recognised in section IV-A, in order to select a diverse less-redundant subset that contains as much SCUs as possible. First, `Ml-DfWebQA` computes a template representation of each definition sentence by replacing sequences of words that start with a capital letter with a placeholder. From now on in this section, in order to avoid confusion, these templates will be referred to as descriptive sentences. Second, `Ml-DfWebQA` obtains templates that match these descriptive sentences from the repository. `Ml-DfWebQA` clusters these templates into groups according to their lengths. Let Θ_l be the group containing templates of length l, and $fmax_{\Theta_l}$ the frequency of its highest frequent element. Subsequently, `Ml-DfWebQA` ranks a definition $d \in D$ according to:

$$R_\Theta(d) = \sum_{l=2}^{10} \xi_l \sum_{\forall t \in \Theta_l^d} \frac{freq(t)}{fmax_{\Theta_l}}$$

In plain words, each definition d is ranked according to its matching templates ($\Theta_l^d \subseteq \Theta_l$). This ranking value consists solely of the sum of the respective normalised frequencies (divided by $fmax_{\Theta_l}$) and a weight ξ_l. This weight factor favours definitions that match longer templates. Third, `Ml-DfWebQA` ranks definitions according to their entities. Taking entities into consideration is vital, because entities are defined by their relations with other entities. Here, `Ml-DfWebQA` builds a frequency histogram of numbers and tokens that start with a capital letter. Each definition is then ranked by adding the frequencies of the entities it carries. These ranking values are thereafter normalised by dividing by the highest value. Let $R_E(d)$ be the normalised value corresponding to the definition d.

The reason to avoid Name Entity Recognisers is two-fold: (a) they perform poorly on web snippets, due to truncations, and (b) it is our aim to use as few as possible linguistic tools and knowledge at the time of extracting answers, while at the same time, increasing the off-line linguistic processing while building our models. This way our system could deal, in the future, with additional languages by only changing the content in the repositories.

Fourth, `Ml-DfWebQA` constructs an histogram H of pairs and triples $\vec{w}$ from the descriptive phrases D. Then, it obtains the respective word association ratios from the repository (I_2 and I_3), and normalises these ratios by dividing by the ratio corresponding to the highest pair and triple afterwards ($\bar{I}_2$ and $\bar{I}_3$, respectively). Later, pairs and triples $\vec{w}$ with a frequency equal to one are removed from the histogram H, and this histogram is normalised similarly to the association ratios ($\bar{H}$). Each definition d is subsequently ranked according to the tuples in the repository as follows:

$$R_I(d) = \sum_{\forall \vec{w} \in \vec{W}^d - \tilde{W}} \bar{I}_2^{'}(\vec{w}) + \bar{I}_3(\vec{w})$$

Where $\tilde{W}$ includes all tuples belonging to previously selected phrases, and $\vec{W}^d$ are all the tuples extracted from the definition $d \in \mathcal{D}$. This $\tilde{W}$ assists in ranking definitions according to their novelty respecting the already selected phrases. $\bar{I}_2^{'}(\vec{w})$ is defined as follows:

$$\bar{I}_2^{'}(\vec{w}) = \begin{cases} \bar{I}_2(\vec{w}) & \text{if } \bar{I}_2(\vec{w}) \neq 0 \\ \bar{H}(\vec{w}) & \text{otherwise} \end{cases}$$

Eventually, a sentence is ranked as follows:

$$R(d) = (1 + R_\Theta(d) + R_E(d)) * R_I(d)$$

The higher ranked sentence is selected and its corresponding tuples are added to $\tilde{W}$, this way sentences containing novel and promising tuples are preferred to more redundant sentences, whose ranking values tend to decrease as long as more phrases are selected. Sentences that obtain a rank value lower than $R(d) < \xi$ are unconsidered.

V. Evaluation

A. Evaluation Metric

The first metric widely used for assessing definition QAS in the TREC track was the $\mathcal{F}_\beta$ score [13]. This measure makes a judgement about the output of a system with respect to a gold standard by taking into consideration the next aspects:

v = number of vital nuggets returned in a response.
o = number of okay nuggets returned in a response.
g = number of vital nuggets in the gold standard.
h = number of non-whitespace characters in the whole output.

In this assessment, the gold standard provides a hierarchy of SCUs, which comprises vital nuggets (must be in the description of the concept) and okay nuggets (not necessary). Then, a length allowance (α) of 100 non-whitespace characters per matched nugget was imposed in order to deal efficiently with two crucial aspects: (a) different paraphrases of a particular SCU can be found, and hence their corresponding lengths differ from one rewriting to the other, and (b) many nuggets

need their context to be readily comprehensible. The allowance of the output of a system is accordingly defined as follows:

$$\alpha = 100 \times (v + o)$$

If the length of response exceeds this allowance, the precision (P) obtained by the system is then linearly downgraded:

$$P = \begin{cases} 1 & \text{if h} < \alpha \\ 1 - \frac{h-\alpha}{h} & \text{otherwise} \end{cases}$$

It is worth noting here that descriptive sentences taken from web snippets are about 110 non-whitespace characters long in average [6], and thus they can be interpreted as SCUs. Subsequently, the recall (R) of the system is calculated as follows:

$$R = \frac{v}{g}$$

The recall only rewards a system for the discovered vital nuggets. The $\mathcal{F}_\beta$ value is, eventually, computed by balancing the trade-off between precision and recall:

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

In TREC 2003, β was set to five, but since this value was heavily biased in favour of large responses, it was later decreased to three. In fact, the value of β depends chiefly on the type of application. For example, a system for a cell phone platform would prefer concise SCUs to large responses, while a web search engine would prefer more contextual information.

As [5], [6], [14] duly pointed out, if a system does not discover at least one vital nugget, it finishes with a recall equal to zero, bringing about a $\mathcal{F}_\beta$ score equal to zero, even though the system outputs okay nuggets and their output lengths can dramatically differ. Since these zero values are completely unuseful for comparing systems, [14] proposed a new $\mathcal{F}_\beta$, which modifies the recall to make allowances for weighted nuggets:

$$R = \frac{\sum_{x \in A} z_x}{\sum_{y \in Z} z_y}$$

Where $A \subseteq Z$ is the set of nuggets in the gold standard that matched the response of the system and Z is the set of all SCUs in the gold standard; z_x and z_y are the weights of the nuggets x and y, respectively. [14] computed these weights by averaging the opinions of several assessors regarding the SCUs in the gold standard. In TREC, this gold standard is manually compiled, and it is known that systems in TREC were able to find relevant nuggets, which were not included in this list (cf. [7] for details). In the case of web-based systems, this vital fact is more likely to happen, because systems discover many additional nuggets seen as relevant by the user, but excluded from the TREC gold standard. This exclusion actually brings about a decline in the $\mathcal{F}_\beta$ score, because these extra nuggets enlarge the response without increasing precision.

There are two vital aspects to evaluate in definition question answering systems aimed at the web: (a) the search strategy, and (b) the answer extraction process. The former is crystal clear, a system outperforms another one if it retrieves a larger amount of different nuggets. The latter is the performance in terms of $\mathcal{F}_\beta$ respecting this fetched set of nuggets. In this case, these SCUs are interpreted as the desired output, and thus, as the gold standard. By and large, extracting this gold standard is an arduous task, because it inherently involves manually checking the target corpus. To illustrate, in our work, the TREC 2003 consists of 50 different concepts: 30 are for people (e.g., "*Alberto Tomba*"), 10 are for organisations (e.g., "*ETA*") and 10 are for other entities (e.g., "*vagus nerve*"). For each question, our system retrieves about 300 snippets, therefore 15,000 snippets must be manually checked in order to determine this gold standard. This number doubles to about 30,000, when a baseline system is taken into consideration.

Once the gold standard was determined, we equally weighted SCUs, that is $z_y = 1$, $\forall z_y \in Z$. The reason to use these uniform weights is three-fold: (a) under the assumption that more relevant nuggets will be included in a larger amount of documents, we attempted to weight them according to the number of snippets where they occur, but this causes all systems to obtain a high recall, because high frequent SCUs are usually easier to discover, and little is gained when nuggets low in frequency are detected, (b) if a high frequent SCUs is missed by a system, it needs many low frequent nuggets to recover from the loss, and (c) the gold standard is largely dependent on the search strategy, and therefore the distribution of weights could sharply vary, turning to be an important factor when comparing different systems. All things considered, we define a $\mathcal{F}_\beta^{web}$ score which computes the recall R as $\frac{v'}{g'}$, and α as $100 \times v'$. Where v' and g' are the number of different answers that the system recognises and retrieves, respectively.

B. Experiments

As mentioned in the previous section, `Ml-DfWebQA` was assessed by means of the fifty questions supplied by the TREC 2003 track , and 19 queries respecting the CLEF 2008 Spanish track. In our experiments, we used MSN Search[7] as interface between `Ml-DfWebQA` and the web.

In order to test the efficiency of the presented methods, we used as `Baseline` the system proposed in [6]. Since both systems are aimed at web snippets and share a similar spirit, it is a good starting point for assessing the strategies introduced in this work. More precisely, the essential difference is that `Baseline` attempts to extract answers based of frequency counts, that is without the assistance of external sources of knowledge (e.g., Wikipedia). It is worth noting that both systems fetch about the same number of snippets, and that `Baseline` deals also with definition queries in Spanish.

Table V remarks the value of parameters utilised in our experiments. ξ_l were fixed according to the number of matching templates across a subset of the English CLEF definition questions. Certainly, longer templates are more reliable and harder to match, and thus they are weighted more heavily. Several

[7] http://www.live.com/

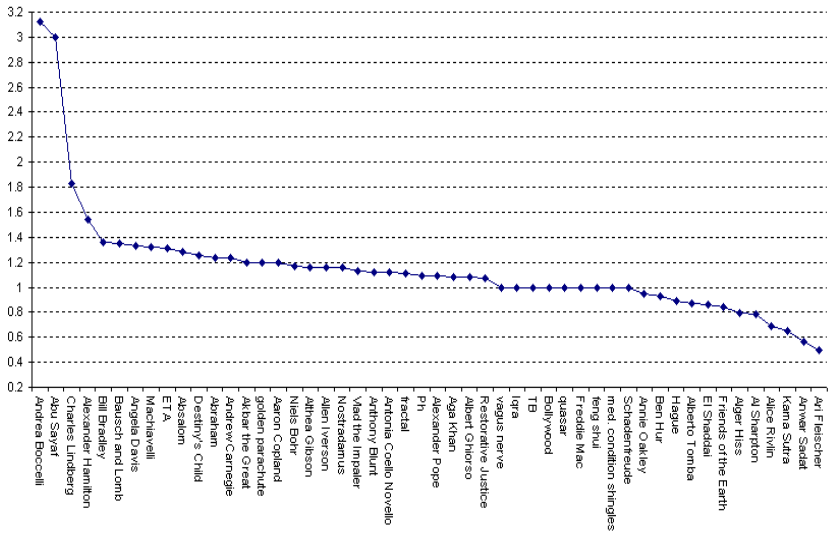

Fig. 1. $\frac{g'_{\texttt{Ml-DfWebQA}}}{g'_{\texttt{Baseline}}}$ vs. *definiendum*.

TABLE V

ML-DFWEBQA PARAMETERS.

ξ_2	ξ_3	ξ_4	ξ_5	ξ_6
0.0528	0.0708	0.0861	0.09407	0.09759
ξ_7	ξ_8	ξ_9	ξ_{10}	ξ
0.09916	0.09955	0.09983	0.09989	0.1

TABLE VI

TREC 2003 RESULTS ($\mathcal{F}_\beta^{web}$ SCORE).

	Baseline	Ml-DfWebQA
$\mathcal{F}_1^{web}$	0.44 $\pm$ 0.16	0.42 $\pm$ 0.14
$\mathcal{F}_2^{web}$	0.44 $\pm$ 0.16	0.48 $\pm$ 0.13
$\mathcal{F}_3^{web}$	0.45 $\pm$ 0.17	0.51 $\pm$ 0.14
$\mathcal{F}_4^{web}$	0.45 $\pm$ 0.17	0.53 $\pm$ 0.15
$\mathcal{F}_5^{web}$	0.46 $\pm$ 0.18	0.54 $\pm$ 0.16

ξ values were tried (0 to 0.3) to optimise Ml-DfWebQA by using the same subset of questions. As a rule of thumb, values higher than 0.3 can miss many novel nuggets.

Figure 1 highlights the ratio of the number of SCUs fetched by Ml-DfWebQA ($g'_{\texttt{Ml-DfWebQA}}$) to the nuggets retrieved by the Baseline ($g'_{\texttt{Baseline}}$). The average value[8] of this ratio was 1.15 $\pm$ 0.46. This improvement is due to 29 questions (58%), for which Ml-DfWebQA retrieved a higher number of different nuggets, whereas in twelve cases (24%) Baseline fetched more nuggets. In nine (18%) of the questions, there was not tangible enhancement or decrease. The interesting point in figure 1 is that the three more remarkable improvements are due to our strategy of finding alternative aliases. In the best case, the original concept is "*Andrea Boccelli*", but Ml-DfWebQA found out that "*Andrea Bocelli*" is a better writing. Given these outcomes, it can be concluded that our repository of aliases is especially helpful for the robustness of this class of systems. However, we envision that, whenever it is possible, an intermediate phase consisting of requesting at the user for the validation of the aliases instead of querying the web, would be more appropiate. In the fourth top *definiendum*, the following two queries boosted the recall of descriptive phrases:

q_2: "Alexander Hamilton, who wrote" $\vee$ "Alexander Hamilton that resulted in" $\vee$ "Alexander Hamilton, who favored" $\vee$ "Alexander Hamilton who served at"

q_5: "Alexander Hamilton was born in" $\vee$ "Alexander Hamilton was born on" $\vee$ "Alexander Hamilton , a founding" $\vee$ "Alexander Hamilton who served at"

On the other hand, in the case of "*Ari Fleischer*", the decrease in performance was due to selected clauses that were semantically similar, and hence, they brought about the retrieval of descriptive phrases that convey similar SCUs:

q_5: "Ari Fleischer , the president" $\vee$ "Ari Fleischer , the press" $\vee$ "Ari Fleischer , a spokesman"

Figure 2 plots the ratio of the $\mathcal{F}_\beta^{web}$ corresponding to both systems. Ml-DfWebQA outperformed Baseline in 34 questions(68%), whereas Baseline finished with a higher score for 16 questions (32%). First of all, there was no profound difference in the results per question between $\beta = 3$ and $\beta = 5$. In 13 questions, Ml-DfWebQA obtained more than 50% improvement, while in 17 more than 30% and in 27 more than 20%. On the other hand, the performance was considerably decreased in 10 cases (20%). Given these results, we can conclude that the presented strategies help to distinguish more SCUs low in frequency.

There are two decisive factors which worsen the perfor-

[8]Along this section, $\pm$ stands for standard deviation.

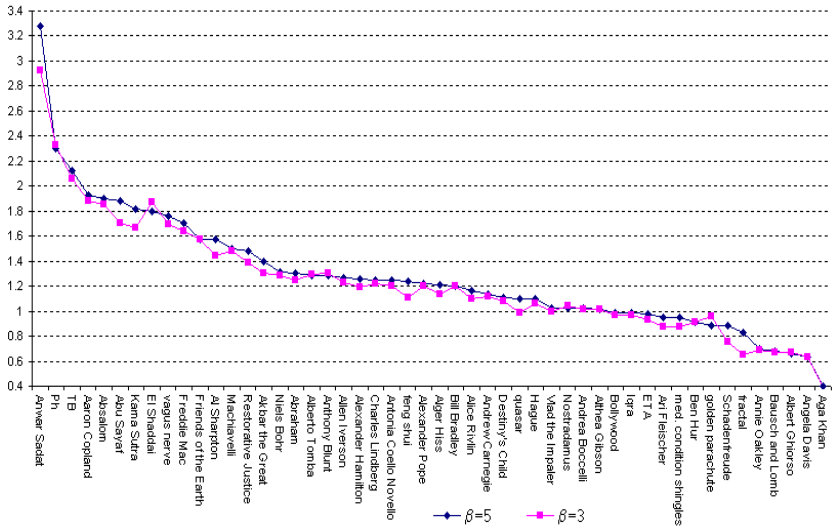

Fig. 2. $\frac{\mathcal{F}^{web}_{\texttt{Ml-DfWebQA}}}{\mathcal{F}^{web}_{\texttt{Baseline}}}$ vs. *definiendum*.

mance of Ml-DfWebQA. First, paraphrases that do not share a substantial number of words, but basically convey (almost) the same descriptive information:

Schadenfreude is a German word that refers to the guilty joy people sometimes feel at the misfortune of others.
Schadenfreude is a German word meaning to take pleasure at the misfortune of others.

The second determining factor derives from the first: two sentences that share many words, but the few changed terms bring about many significant tuples, and Ml-DfWebQA, therefore, interpret this sentence as carrying significant novel definition information. To illustrate this, consider the following two selected definitions:

Schadenfreude is a German word that means "pleasure derived from the misfortunes of others".
Schadenfreude is a German word meaning to take pleasure at the misfortune of others.

The change, here, of "*that means*" to "*meaning*" causes the matching of the tuples <meaning, taken, 3.93> and <means, taken, 3.86>. Both carry the same meaning but they are seen differently by Ml-DfWebQA.

In the light of the obtained results, it can be concluded that our templates and tuples[9] can assist in bettering the efficiency and robustness of definition QASs in English. However, these results cannot be extended to Spanish. Figure 3 shows the outcome for 19 CLEF definition questions ($\beta = 3$). The performance was improved in six questions, whereas diminished in nine cases. The reason to this decrease is that Wikipedia supplies about 2,000,000 definition pages in English, while about 200,000 in Spanish. Therefore, the association ratios computed for Spanish were not as reliable as for English. Additionally,

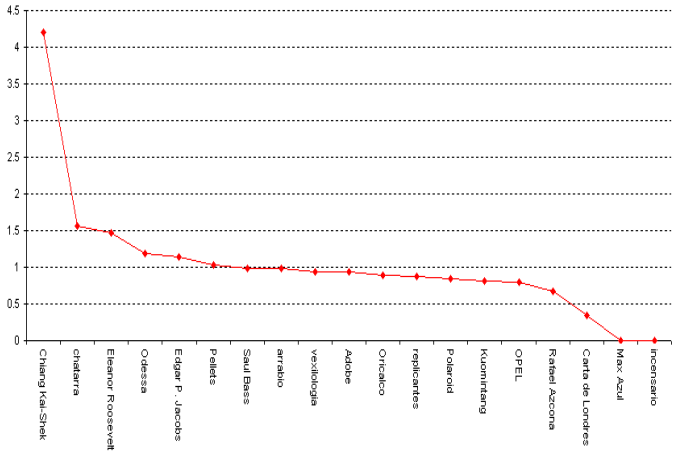

Fig. 3. $\frac{\mathcal{F}^{web}_{\texttt{Ml-DfWebQA}}}{\mathcal{F}^{web}_{\texttt{Baseline}}}$ vs. *definiendum*.

the number of tuples in English extracted from Wikipedia is (at least) three times larger than in Spanish. Therefore, it is harder to find matches within web snippets. Figure 4 corroborates this factor. In this figure, the performance betters, that is $\mathcal{F}^{web}_{\beta}$ closer to one, when a higher number of triples was matched (Zone II), whereas it worsens when few matching triples were discovered (Zone III). Eventually, table VII shows the $\mathcal{F}^{web}_{\beta}$ score for the CLEF 2008 question set.

ACKNOWLEDGMENT

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.

Ml-DfWebQA can be accessed online[10], and the user can

[9] Available under http://www.dfki.de/~figueroa/

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

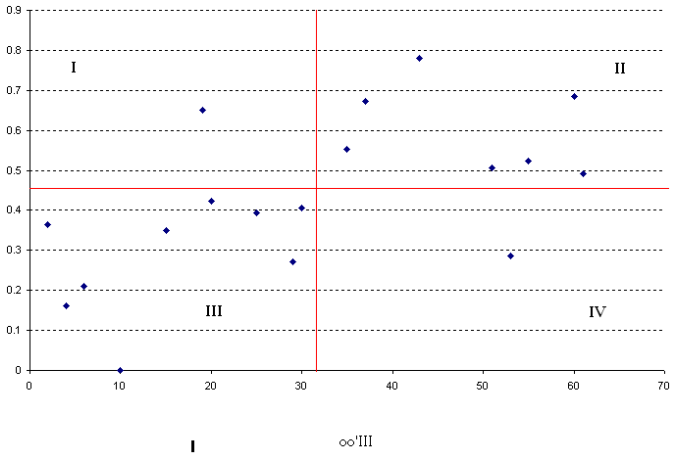

Fig. 4. $\mathcal{F}^{web}_{\texttt{Ml-DfWebQA}}$ vs. Number of Matched Triples.

TABLE VII
CLEF 2008 RESULTS ($\mathcal{F}^{web}_{\beta}$ SCORE).

	Baseline	Ml-DfWebQA
$\mathcal{F}^{web}_{1}$	0.37 ± 0.18	0.40 ± 0.22
$\mathcal{F}^{web}_{2}$	0.43 ± 0.20	0.43 ± 0.23
$\mathcal{F}^{web}_{3}$	0.47 ± 0.22	0.44 ± 0.24
$\mathcal{F}^{web}_{4}$	0.48 ± 0.23	0.44 ± 0.24
$\mathcal{F}^{web}_{5}$	0.50 ± 0.24	0.45 ± 0.24

call our definition module by means of the feature "define:*definiendum*". By default, `Ml-DfWebQA` searches for definitions in English, however it can be switched to Spanish by specifying "language:es".

VI. CONCLUSIONS AND FUTURE WORK

The strategies proposed in this work improve the efficiency and robustness of multilingual definition QASs. However, more definition resources are necessary for languages other than English like Spanish.

For future work, we envisage using dependency parsing for improving our models, this way more accurate association ratios can be computed, leading to a better ranking of sentences.

REFERENCES

[1] D. E. Rose and D. Levinson, "Understanding user goals in web search," in *WWW '04: Proceedings of the 13th international conference on World Wide Web*, 2004, pp. 13–19.

[2] T. 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.

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

[4] J. Xu, Y. Cao, H. Li, and M. Zhao, "Ranking definitions with supervised learning methods," in *WWW2005*, 2005, pp. 811–819.

[5] A. Figueroa, "Boosting the recall of descriptive phrases in web snippets," in *In LangTech 2008*, 2008.

[6] A. Figueroa and G. Neumann, "A Multilingual Framework for Searching Definitions on Web Snippets," in *KI*, 2007, pp. 144–159.

[7] W. Hildebrandt, B. Katz, and J. Lin, "Answering Definition Questions Using Multiple Knowledge Sources," in *HLT-NAACL*, 2004, pp. 49–56.

[8] H. Joho, M. Sanderson, "Retrieving Descriptive Phrases from Large Amounts of Free Text," in *9th ACM conference on Information and Knowledge Management*, 2000, pp. 180–186.

[9] H. Joho and M. Sanderson, "Large Scale Testing of a Descriptive Phrase Finder," in *1st Human Language Technology Conference*, 2001, pp. 219–221.

[10] A. Téllez, A. Juárez, G. Hernández, C. Denicia, E. Villatoro, M. Montes and L. Villaseñor, "INAOEs Participation at QA@CLEF 2007," in *Working Notes for the CLEF 2007 Workshop*, 2007.

[11] M. Montes–y–Gómez, L. Villaseñor-Pineda, M. Pérez-Coutiño, J. M. Gómez- Soriano, E. Sanchis-Arnal and P. Rosso, "INAOE-UPV Joint Participation in CLEF 2005: Experiments in Monolingual Question Answering," in *Working Notes for the CLEF 2005 Workshop*, 2005.

[12] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational Linguistics*, vol. 16, no. 1, pp. 22–29, 1990.

[13] E. M. Voorhees, "Overview of the trec 2003 question answering track," in *Proceedings of TREC 2003*, 2003.

[14] J. Lin and D. Demner-Fushman, "Will pyramids built of nuggets topple over?" in *Proceedings of the main conference on HTL/NAACL*, 2006, pp. 383–390.