Seeking Acronym Definitions: a Web-based Approach

David SÁNCHEZ, David ISERN
Universitat Rovira i Virgili
Departament d'Enginyeria Informàtica i Matemàtiques
ITAKA Research Group
Avda. Països Catalans, 26. 43007 Tarragona, Catalunya
david.sanchez@urv.cat, david.isern@urv.cat

Abstract. Acronyms are widely used in many domains to abbreviate and stress important concepts. Due to its dynamicity and unbounded nature, manual attempts to compose a global scale repository of acronym-definition pairs result in an overwhelming amount of work and limited amount of results. Attending these shortcomings, the paper presents an automatic and non-supervised methodology to generate acronyms and extract their possible definitions from the Web. The method has been designed in order to minimize the set of constraints, offering a domain and -partially- language independent solution. The obtained results have been manually evaluated against the largest manually built acronym repository (Acronym Finder). The results obtained after this comparison show that the proposed automatic web-based approach is able to improve the coverage of manual attempts offering a high precision.

Keywords. Acronyms, information extraction, Web mining.

1. Introduction

Acronyms are common textual forms used to refer to relevant concepts or entities. Human languages are very prone to the creation of acronyms in order to *i)* stress the importance of entities , *ii)* avoid redundancy by omitting entity's long form and *iii)* offer an alternative form for referring to the same entity which is easier to remember. Acronyms comprise a very dynamic domain, as new acronyms are defined every day for any possible domain of knowledge. They are highly polysemic because they are composed by short combinations of characters (typically from 2 to 6). Finally, they have a very diverse degree of generality with very common ones (*e.g.,* USA - United States of America), and others appearing very rarely (*e.g.,* USA - Unique Settable Attributes).

Formally, an acronym may correspond to one or more *definition(s)* (the full long form of the referred entity) from which several *participating* characters are used to construct the acronym.

Identifying equivalent acronym-definition pairs is a crucial task in natural language processing and information retrieval tasks [4,11]. Ontology Population and Questions Answering are other areas in which adequate acronym handling can bring benefits [15]. However, due to the previously introduced characteristics, it is very difficult to construct a general and up-to-date database of acronym-definition pairs [14]. From the manual point of view, there have been some ambitious attempts to provide global and reliable acronym dictionaries such as Acronym Finder[1] or The Internet Acronym Server[2]. Those repositories offer a valuable source of knowledge at the cost of a high amount of human work. More specifically, Acronym Finder, which is the world's largest dictionary of acronyms, has more than 750,000 human-edited definitions; the total effort required to compile this set is estimated to be more than 6,000 hours[3]. Moreover, the knowledge acquisition bottleneck introduces serious limitations about coverage, as it will be shown in the evaluation section. Consequently, automated methodologies may aid the composition of a general and up-to-date acronym-definition repository.

Recently, there have been some approaches focused on the automatic side to identify acronym-definition pairs within text. Most approaches use a set of patterns to extract definition candidates for a given acronym and some heuristics (mainly constraint sets) to evaluate its suitability. The use of capitalization heuristics [6] (*i.e.*, acronyms should represent be presented with capitalized words), letter matching [10] (*i.e.*, the first(s) letter(s) of the acronym definition should match acronym's letters) and parenthetical rules [13] (*i.e.*, acronym and definitions are adjacent terms represented within parenthesis) are the most commonly used and effective ones [7]. Some of them [6,9,13] employ a certain degree of linguistic analysis (*e.g.*, detection of stop words, use of language related linguistic patterns) which hampers its applicability to other languages.

In non-supervised approaches, the result's quality depends highly on the set of constraints used to filter acronym definition candidates. As stated in [7], the use of strong constraints over a reduced corpus (such as in [15]) results in high precision but compromises the final recall. In order to minimize this problem, some authors [2,7] designed supervised approaches, requiring a certain amount of manually tagged data and training.

These approaches are applied over English written resources due to the introduced language related constraints, being very few of them multi-language [15]. Some authors have dealt with other languages, such as Japanese [5] or Korean [14].

The mentioned works use a document or a set of documents as source. This is an implicit way of contextualizing the search towards a certain domain, however, this introduces limitations as, if no explicit matching for the defined patterns appears within the text, no results will be obtained. Very few approaches [8,14] have been developed to create a large scale acronym dictionaries.

Considering the lack of approaches dealing with the creation of large acronym dictionaries, this paper presents a novel method to, at first, generate valid acronyms, and then, retrieve feasible definitions. The proposal uses the Web as source and handles these tasks automatically and in an unsupervised way, creating large acronym-definition sets that can aid to extend manually-composed repositories. Internally, it is based on general extraction patterns and a reduced filtering constraint set in order to configure a domain-independent approach. As no language dependant analyses are introduced, potentially, cross-language results can be retrieved. However, the presented

[1] Web site: http://www.acronymfinder.com/ [Last access: 18/05/2009]

[2] Web site: http://silmaril.ie/cgi-bin/uncgi/acronyms. [last access: 14/05/2009]

[3] Source: http://blog.acronymfinder.com/2008/04/acronym-finder-rolls-over-600000.html.[Last access: 18/05/09]

proposal is limited to languages written with Latin characters (*e.g.*, English, Catalan, Italian, etc). Results have been manually evaluated against a human-made repository in order to check their accuracy and to show the benefits that an automatic approach can bring in comparison to manual attempts.

The rest of the paper is organized as follows. Section 2 describes in detail the proposed methodology to create a general acronym-definition repository. Section 3 introduces the evaluation process; it shows the obtained results and presents an analysis of them. The last section presents the conclusions and proposes some lines of future work.

2. Extraction of acronym-definition pairs from the Web

The proposed methodology is divided in several parts. At first, a simple algorithm generates possible acronyms by the combination of alpha-numeric characters of a specific length. Then, for all those generated candidates, the system tries to discover all possible definitions in the Web. It retrieves as many acronym definitions as possible trough several executions of the analysis process. Finally, the list of definitions found is filtered using a set of general rules.

2.1. Generating acronyms

Our generation algorithm creates sets of strings combining Latin letters (A-Z) and numbers (0-9) of a given length (*e.g.*, URV, UPC, UDL, and I2N, in the case of 3 letters lengthen acronyms).

Candidates composed only by numbers are avoided because they are not acronyms. Other possibilities containing special characters (*e.g.*, ".","-") are not considered because these characters are not included as participant letters by Web search engines (*e.g.*, R.A.D.A.R. will lead to the same results than RADAR being the two forms a valid acronym for *RAdio Detection And Ranging*).

2.2. Retrieving definitions

For each generated acronym, the system queries a web search engine in order to retrieve a corpus of documents containing acronym matches. With the set of resources, an analysis to extract definition candidates is performed. For efficiency purposes and, considering that each website is typically using the acronym in a concrete sense [12], we only analyse the Web abstracts (snippets) provided by the web search engine, which present two or three lines in which the query matching appears. Two advantages arise from this approach: *i)* the minimization of the number web accesses which puts a high computational burden on the execution (with only one query we are able to retrieve up to 200 different web snippets), and *ii)* the heterogeneity of the corpus, maximizing the diversity of the information sources and, consequently, the acronym senses in polysemic cases.

The snippet set is scanned according to four general acronym-definition patterns (shown Table 1). We only consider patterns involving punctuation symbols widely used in Latin-based languages, which have been demonstrated to offer the best performance [7] at the cost of being relatively restrictive. However, considering the amount of resources available and the information redundancy which characterizes the

Web [1], data sparseness derived from constrained pattern matching will be minimised (*i.e.* the same information can be found several times in many different expressive forms).

Table 1: List of language-independent acronym-definition extraction patterns

Pattern	Example
Acronym (definition)	NBC (National Broadcasting Company)
Definition (acronym)	British Broadcasting Corporation (BBC)
Acronym -definition-	CNN -Cable News Network-
Definition –acronym-	Home Box Office -HBO-

The list of definition candidates extracted for a given acronym is then filtered in order to select only those which fulfil a set of rules (see Table 2). We have defined them to be general (*i.e.* they are domain independent) and applicable for different languages (*i.e.* they do not involve language-dependant linguistic constructions).

The stronger constraints are established by the first two ones, which are meant to discover typical acronym constructions (*i.e.* the acronym is composed by several ordered *participant* letters of the definition). Rule 3 may also introduce compromises as it is possible to lose some meaningless words belonging to the definition (*e.g.*, determinants). However, we need to use this constraint to detect the beginning of a definition in patterns like *definition(acronym)* or *definition –acronym-* without relying on language dependant linguistic analysis (*i.e.*, detection of stop-words). In those cases the end of the definition is detected using the punctuation symbol ('(' or '-'). Consequently, we take the longest possible definition fulfilling the specified rule. Rules 4 and 5 are introduced to reduce the possibility of retrieving malformed candidates (*e.g.*, missing terms in the definition or arbitrarily concatenated text). Those cases are very common when dealing with snippets, as web abstracts' text length is more or less arbitrarily.

Table 2: Acronym definition filtering rules

Rule	Description
$Rule_1$	All acronym characters must appear in the definition
$Rule_2$	Acronym characters must appear in the same order as in the definition
$Rule_3$	Definition must begin with the same letter as the acronym
$Rule_4$	Definition maximum length is n*10, where n is the number of acronym characters
$Rule_5$	Definition must have at least one more character than the acronym

Compared to previous approaches, other authors (working on corpuses with limited scope) introduce stronger rules in order to improve the precision at the cost of a lower recall (like "*only the first letter of the definition can participate*" [10] or "*first three letters of the definition can participate*" [13]).

2.3. Expanding the analysis

Previously, we mentioned that, for a given acronym and query, a set of web snippets are retrieved and parsed for pattern matches. However, how big should be this set in order to properly acquire a relevant set (with regards to recall) of acronym definitions? On the one hand, due to the automatic nature of the acronym generation process, there will be many occasions in which a combination of characters has never been used as a definition abbreviation; consequently the unproductive analysis of hundreds of resources will hamper the performance. On the other hand, some character combinations may result in highly polysemic acronyms for which many definitions are available and should be discovered; in this case, even a thousand of resources (*i.e*. the maximum result set indexed by a search engine) may be not enough to retrieve some of the definitions.

In order to establish the size of the web corpus to analyse in function on the acronym, we introduce an adaptive algorithm which dynamically increases the corpus size as far as the learning throughput is above a certain mark. So, invalid candidates will be dropped when no definitions are found, meanwhile highly productive ones will be extensively analysed.

Concretely, the algorithm begins analysing the first set of snippets (*e.g.*, 200) for an acronym candidate. Then, the system takes into consideration the amount of new candidate definitions found during this stage, and if it surpasses a minimum threshold (for instance one or two definitions) it continues the analysis by retrieving an additional set of web snippets. The process continues until the new results obtained do not reach the threshold.

This procedure has a practical limitation: even though an acronym queried in a web search engine may return millions of results, only the first 1000 web sites will be indexed (this is a common limitation to general purpose web search engines like Google). In addition, Web search engines typically introduce a relevance-based bias to the list of results. Consequently, only the most general definitions can be retrieved, and rare definitions remain hidden.

In order to overcome recall related problems and attenuate the Web search engine dependency in the results, it is possible to reformulate the initial query to force the Web search engine to retrieve a different set of resources [3].

The specially designed query expansion algorithm relies on previously acquired knowledge to reformulate the query in a smart manner. Concretely, if the full set of 1000 web snippets have been analysed for an acronym, this means that a minimum set of definitions have been retrieved. As a result, we would want that an additional set of web resources to explore do not contain redundant acronym definitions (*i.e.,* use the acronym in the same sense).

In order to avoid retrieving repeated resources or those which cover the same acronym definition, increasing the heterogeneity of the analysed corpus and minimizing the search engine ranking bias, we can add a exclusion restriction to the query using the operators "-" or "NOT". So, the query will be reformulated, including the initial acronym and excluding all the already retrieved definitions (*e.g.*, USA – "United States of America").

However, we observed that most search engines only use exclusion operators to filter the set of resources, rather than presenting a completely new set of resources. In order to force an increase of the result's variance, we introduce an additional heuristic. We observed that it is very common that one or several acronym characters derive from the same word. For example, for the "URV" acronym, the "U" character stands in many occasions to adjectives such as "universal", "unified" or "uniform". On the other hand, the "V" character corresponds in commonly to the noun "Vehicle" or "Value". This uniformity gives a clue that it is likely to discover new definitions using the same terms for which the same acronym has been defined. So, the second query expansion heuristic consists of creating a new set of queries including, in each one, in addition to the acronym and the excluded definitions, a new word which have appeared several times (more than one) in the already acquired definitions. This will provide an additional set of 1000 resources at the maximum for each multi-appeared term.

As an example of the effectiveness of the query expansion algorithm, for the URV acronym, we were able to obtain 13 different definitions analysing the first 1000 snippets. Omitting those ones and including frequently retrieved words such as "Underwater", "Unit", "Value" or "Vehicle", we found 20 new definitions from the analysis of an additional set of 5000 snippets.

3. Evaluation

The evaluation of automatic learning procedures which deal with highly dynamic environments like acronyms and unbounded corpus as the Web is certainly a challenging task. Fortunately, there exist general purpose manually composed acronym-definition repositories, being the mentioned Acronym Finder the biggest and most updated one.

So, taking Acronym Finder as baseline, we evaluate by hand a random set of 3 letters lengthen acronyms. They constitute an especially problematic set because, on the one hand, the amount of available definitions for the possible and potentially productive combinations can be quite overwhelming for manually constructed repositories (unlike shorter acronyms which can be dealt by manual attempts). On the other hand, due to their shortness, they are very polysemic (unlike longer acronyms for which, due to their variability and complexity, a much more reduced amount of definitions is available).

The combination of 3 Latin letters constitutes an acronym candidate set of 17576 possible acronyms to explore. After the presented algorithm is executed over those acronyms using Google, a list of a least one definition has been retrieved for the 70% of candidates. This percentage indicates that combinations of three characters constitute an especially productive acronym set.

Due to the impossibility of manually evaluating the full set of definitions (representing a set of hundreds of thousands), we took a random subset of 20 acronyms with 5 or more definitions available in Acronym Finder to force the evaluation of polysemic cases (see Table 3).

A definition is considered as correct if it appears on the Acronym Finder list. In the case it is not present, an expert is requested to test the validity of the definition to finally decide its correctness and compute the algorithm's *precision*. The total number of acronym-definition pairs manually evaluated is 1687. *Recall* is computed considering as a baseline the set of definitions presented by Acronym Finder for a given acronym. Finally, *F-measure* is computed as the harmonic mean of precision and recall.

As a result, we obtained a high precision among 85-96%. This high accuracy shows the effectiveness of patterns used to extract candidates and the rules employed to filter them. Even though dealing with different corpus, this precision is higher than previous works attempting to compose large scale acronym-definition sets (like in [8] which a precision of 78% is reported).

Regarding the result's recall, it is lower and more variable, even though maintained at a usable range of 53-78%. In order to study the causes of this situation, we analysed other indicators. First, the absolute number of definitions automatically retrieved is much higher (almost one order of magnitude) than the list presented by

Acronym Finder. Considering that the additional definitions have been validated by an expert, we can see the coverage limitations presented by manually constructed repositories.

Table 3: Evaluation results for 20 three-lettered acronyms against Acronym Finder

Acronym	#Definitions AcroFinder	#Retrieved Definitions	Precision	Recall	F-measure
ABG	19	115	94%	52.6%	67.4%
CNL	16	87	91%	50%	64.5%
ETN	10	57	89.4%	70%	78.5%
IQC	8	28	92.8%	75%	82.9%
IQL	5	13	92.3%	80%	85.7%
KMP	9	90	86.6%	77.7%	81.9%
LEF	15	111	95.5%	60%	73.7%
NIO	7	46	86.9%	71.4%	78.4%
NLE	14	38	94.7%	57.1%	71.2%
NRF	20	111	96.4%	80%	87.4%
OLT	20	110	95.4%	75%	84%
RBN	9	54	92.6%	66.6%	77.5%
SFE	19	177	92.6%	63.1%	75%
TWF	13	87	93.1%	69.2%	79.5%
TWI	13	101	85.1%	69.2%	76.3%
VDC	17	134	95.5%	70.6%	81.2%
VSW	9	30	90%	66.6%	76.5%
WME	13	97	89.6%	69.2%	78.1%
WRP	14	151	96.7%	71.4%	82.1.%
WSN	13	50	96%	53.8%	68.9%

Next, we analysed against the Web the Acronym Finder definitions which the system was not able to discover. First, we discovered that the low recall was not caused by our filtering rules because most of the missing definitions fulfil them. In order to analyse other causes, each non-retrieved definition was queried in conjunction with the acronym into the web search engine to estimate the number of available Web documents for which an explicit acronym-definition matching can be extracted. As a result we found that only a 25% of the performed queries returned more than 10 results. From the remaining 75%, a significant 20% of the definitions returned zero hits. So, one can observe that the missing definitions correspond in their major part to rare definitions with very low amount of Web occurrences (at least indexed by the web search engine). This shows that recall problems are associated to the rarest definitions. Recall limitations have been also observed in previous non-supervised works attempting to construct acronym dictionaries (such as [14] with a maximum recall of 70,9%).

In addition to the general evaluation, we also tested the influence of query expansion algorithm described in section 2.3. We compared the results obtained when analysing the static list of web resources presented by the search engine when querying the acronym (*i.e.*, no query expansion, only 1000 web sites available) against the results obtained when applying the adaptive corpus analysis.

As main conclusion, we found that in all the tested cases, the 1000 directly accessible web resources provided by the search engine were not enough to obtain a representative set of definitions. From the average amount of 101 definitions per acronym retrieved by means of the query expansion algorithm, only an average of 11 is achieved with the 1000 directly accessible ones. This results in a much lower recall with an average of only 21,76% compared against the 68,3% obtained after the initial query is expanded. In both cases, the precisions obtained were very similar (91,72% *vs* 92,31%). This shows the necessity and the usefulness of the incremental query expansion algorithm in order to obtain results with good coverage and overcome the limitations of web search engines.

As a final test, we have evaluated the performance of our approach for the most common situation: the case in which an acronym result in a reduced set of definitions. Long combinations of characters (longer than 3 letters) typically result in a limited amount of definitions. As in those cases the acronym clearly identifies its definition (*i.e.*, the amount of polysemy is minimum or non-existent) our approach will deal with the simplest situation from the Information Extraction point of view: the search engine will immediately provide the resources covering matching of the exact definitions to extract.

In this case, we take another random sample of 20 acronyms with 4 letters for which Acronym Finder provides a minimum of 1 definition and a maximum of 5. We obtained 159 acronym-definition pairs, which they have been manually evaluated. In this scenario, the recall was maximum in most situations, resulting in an average of 90,4%. Precision follows the same tendency observed in previous tests, showing a high accuracy (94% in average) in most situations.

4. Conclusions and future work

In this paper we introduced a new approach aimed to extract acronym-definitions pairs using the Web as a corpus. The amount of information available in the Web and its redundancy is exploited to improve the low recall which typically affects non-supervised approaches [7]. Considering the size of the Web, the enormous amount of possible acronyms and definitions and the non-scalability of supervised attempts, the approach is completely automatic. The generality of the algorithm, based only on general patterns and a reduced constraint set and the exploitation of the Web by means of an incremental and adaptive corpus analysis are the main contributions which differentiates our approach from other works.

Moreover, considering that most of the previous attempts dealing with acronyms are focused in the contextualized detection and discovery of acronyms and definitions in a document, our approach can also contribute by offering a more general solution.

The designed non-supervised method has proven to offer accurate results (after the manual evaluation of more than 1600 randomly selected acronym-definition pairs) with a reasonable level of coverage in comparison to human-made repositories. The number of results obtained is an order of magnitude bigger than manual attempts, which show the usefulness of the proposal. Specifically, it may aid in the development this kind of manually composed repositories, improving the recall and maintaining the results up-to-date.

With the current results, however, we have no clue about the generality or relatedness of each definition with respect to the input acronym, as they have been extracted from individual observations. As a future line of research, we plan to add a generality-based definition rating algorithm in order to rank the list of extracted definitions for each acronym according to their degree of relatedness computed form the Web. Repositories such as Acronym Finder provide this kind of generality-based rankings which aim to ease the search and understanding of definition sets for polysemic acronyms. This can also stand as a baseline to which compare and evaluate automatically obtained ratings.

Another research line consists of the combination of several search engines (*e.g.*, Google, Altavista, MSNLive!) in order to compile a more complete and heterogeneous corpus to analyse. The objective is to overcome the coverage issues for some rare cases as far as the web search engine indexing arrives. Other long term research lines may include the detection of definition language using automatic language recognizers or the automatic clustering of domain related definitions according to, for example, predefined general categories.

Acknowledgements

This work has been supported by the *Departament d'Innovació, Universitats i Empresa de la Generalitat de Catalunya i del Fons Social Europeu* and *Universitat Rovira i Virgili*.

References

[1] E. Brill, Processing Natural Language without Natural Language Processing, in: *4th International Conference on Computational Linguistics and Intelligent Text Processing, CICLing 2003*, A. Gelbukh, ed, Springer Berlin / Heidelberg, Mexico City, Mexico, 2003, 360–369.

[2] J. T. Chang, H. Schütze, Abbreviations in biomedical text, in: *Text Mining for Biology and Biomedicine*, S. Ananiadou, J. McNaught, eds, Artech House, Norwood, US, 2006, 99-119.

[3] P.-A. Chirita, C. S. Firan, W. Nejdl, Personalized Query Expansion for the Web, in: *30th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR 07*, C. L. A. Clarke, N. Fuhr, N. Kando, W. Kraaij, A. P. de Vries, eds, ACM, Amsterdam, The Netherlands 2007, 7-14.

[4] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, A. Yates, Unsupervised named-entity extraction from the Web: An experimental study, *Artificial Intelligence* **165** (1) (2005), 91-134.

[5] T. Hisamitsu, Y. Niwa, Extracting useful terms from parenthetical expression by combining simple rules and statistical measures: A comparative evaluation of bigram statistics, in: *Recent Advances in Computational Terminology*, D. Bourigault, J. Christian, M.-C. L'Homme, eds, John Benjamins Publishing Company, 2001, 209-224.

[6] L. Larkey, P. Ogilvie, A. Price, B. Tamilio, Acrophile: An Automated Acronym Extractor and Server, in: *5th ACM conference on Digital libraries*, Association for Computing Machinery, San Antonio, Texas, US, 2000, 205-214.

[7] D. Nadeau, P. D. Turney, A Supervised Learning Approach to Acronym Identification, in: *18th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2005*, B. Kégl, G. Lapalme, eds, Springer Berlin / Heidelberg, Victoria, Canada, 2005, 319-329.

[8] N. Okazaki, S. Ananiadou, A Term Recognition Approach to Acronym Recognition, in: *International Committee on Computational Linguistics and the Association for Computational Linguistics, COLING-ACL 2006*, Association for Computational Linguistics, Sydney, Australia, 2006, 643-650.

[9] Y. Park, R. J. Byrd, Hybrid Text Mining for Finding Abbreviations and Their Definitions, in: *Conference on Empirical Methods in Natural Language Processing, EMNLP 2001*, L. Lee, D. Harman, eds, Intelligent Information Systems Institute, Pittsburgh, US, 2001, 126-133.

[10] K. Taghva, J. Gilbreth, Recognizing acronyms and their definitions, *International Journal on Document Analysis and Recognition* **1** (4) (1999), 191-198.

[11] L. Xiao, D. Wissmann, M. Brown, S. Jablonski, Information Extraction from the Web: System and Techniques *Applied Intelligence* **21** (2) (2004), 195-224.

[12] D. Yarowsky, Unsupervised Word-Sense Disambiguation Rivaling Supervised Methods, in: *33rd Annual Meeting of the Association for Computational Linguistics*, H. Uszkoreit, ed, Association for Computational Linguistics, Cambridge, MA, US, 1995, 189-196.

[13] S. Yeates, Automatic extraction of acronyms from text, in: *Third New Zealand Computer Science Research Students' Conference*, S. Yeates, ed, University of Waikato, Te Kohinga Marama Marae, Hamilton, New Zealand, 1999, 117-124.
[14] Y.-C. Yoon, S.-Y. Park, Y.-I. Song, H.-C. Rim, D.-W. Rhee, Automatic Acronym Dictionary Construction Based on Acronym Generation Types, *IEICE Transactions on Information and Systems* **E91-D** (5) (2008), 1584-1587.
[15] M. Zahariev. 1991. A (Acronyms). Faculty of Control Systems and Computers, Polytechnic Institute of Bucharest. Bucharest, Rumania: Simon Fraser University.