A set of data bases to support intelligent agents in Internet Infrastructure routing domains

Juliao Braga[1,2], **Joao Nuno Silva**[1], **Nizam Omar**[2]

[1]INESC-ID, Lisbon University, Lisbon, PT

[2]Mackenzie University, Sao Paulo, SP, Brazil

{juliao.braga,joao.n.silva}@tecnico.ulisboa.pt

nizam.omar@mackenzie.br

Abstract. *This paper presents a set of three data bases that make up the Internet Infrastructure Data Base (IIDB). IIDB has three data bases – iidb.rfc, iidb.person, and iidb.acronym – that are key pieces to support the development of machine learning techniques by the intelligent elements of the Autonomous Architecture Over Restricted Domains (A2RD). The data contained in iidb.rfc and iidb.person were created after processing the contents available at the RFC Index web page. While the data contained in the iidb.acronym was created after processing the contents of the files available at the Request for Comments (RFC) repository, produced and maintained by the RFC Editor. The data format of IIDB data is JavaScript Object Notation (JSON), whose templates are available in the same site where the data bases are deposited, making them accessible through any programming language.*

Resumo. *Este artigo apresenta um conjunto de três bases de dados que compõem o Internet Infrastructure Data Base (IIDB). O IIDB é um conjunto formado pelas bases iidb.rfc, iidb.person e iidb.acronym, peças-chave para apoiar o desenvolvimento do aprendizado de máquina desejado aos elementos inteligentes do projeto Arquitetura Autônoma Sobre Domínios Restritos (A2RD). Os dados contidos em iidb.rfc e iidb.person foram criados após o processamento do conteúdo disponível na página web do RFC Index. Enquanto os dados contidos no iidb.acronym foram criados após o processamento do conteúdo dos arquivos disponíveis no repositório Request for Comments (RFC) produzido e mantido pelo RFC Editor. Todos os dados no IIDB são formatados em JavaScript Object Notation (JSON), cujos respectivos modelos estão disponíveis no mesmo site onde as bases de dados são depositadas, acessíveis através de qualquer linguagem de programação.*

1. Introduction

There is a permanent concern to convey enough intelligence to *Autonomous Architecture Over Restricted Domains* (A2RD) agents to make them autonomous. This requires an organized integration of the resources shown in Figure 1, where the A2RD model that can be implemented in each *Autonomous System* AS (or routing domain) [Colel et al. 1994, Hares and Katz 1989], is represented as item (11) in the Figure 1, integrated to the *Structure for Knowledge Acquisition, Use, Learning, and Collaboration* (SKAU) model.

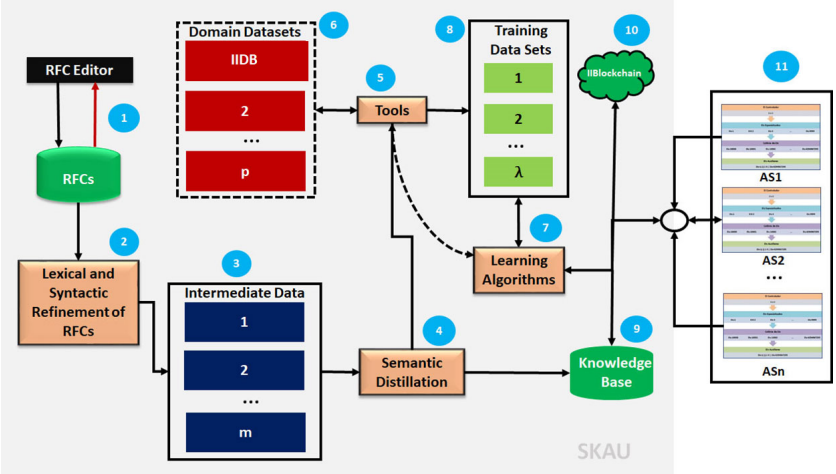


Figure 1. Structure for Knowledge Acquisition, Use, Learning and Collaboration model (SKAU)

The SKAU model components and activities are:

- Each RFC is captured/updated and stored locally and transformed in a corpus, ready to be process by the Python *Natural Language Tool Kit*[1] (NLTK), (1) [Bird et al. 2009];
- A set of tools is responsible for acting lexical and syntactically on RFCs (2), transforming them into intermediary data bases (3);
- Other tools (4), like *Semantic Distillation*, act on the intermediary data bases producing inputs for *Domain Data Sets* (DDS) construction (6) and supporting the data set production that forms the *Training Data Sets* (TDS) (8). Also, these tools will support part of the Knowledge Base (KB) (9) [Isotani and Bittencourt 2015]. A relevant part of DDS is the *Internet Infrastructure Data Base* (IIDB), whose construction description is the goal of this paper;
- *Learning algorithms* (7) support the construction and use of TDS to renew the knowledge base and meet the demand A2RD agents in the process of developing and applied intelligent actions. The efficient use of TDS will respond to the classic algorithms of Machine Learning (ML): *(a)* supervised learning, *(b)* unsupervised learning, *(c)* reinforcement learning and *(d)* semi-supervised learning, that combines *(a)* and *(b)*. [Musumeci et al. 2018] in Section II has an appropriate ML overview, with a focus on optical networks;
- Each implemented A2RD model build a data base, named *IIBlockchain* (10) and stored it together, in the Git Hub (i.e. in the cloud). This then supports the process of collaboration and effective interaction, inter/intra agents of the models [Braga et al. 2018]. The *IIBlockchain* cloud interacts with the *learning algorithm* and KB allowing agents to exercise *offline* and *online computation*[2] [Poole and Mackworth 2010].

Each AS can implement an A2RD, which is then controlled by the IE – *IE Controller* – and receives the identification *x:0*, where *x* is the *AS Number* (ASN).

[1]https://www.nltk.org

[2]*Offline computation is the computation done by the agent before it has to act, and online computation is the computation done by the agent between observing the environment and acting in the environment*

2. IIDB

IIDB evolved from the efforts to build *WordNet* [Fellbaum 1998]. It was later realized that its usefulness would be amplified if it represented not only words (from the domain of the Internet Infrastructure) and their lexical equivalents but also any representation associated with it meaning (proper names, numbers, dates, acronyms, etc.). Table 1 shows an example of the IIDB contents (excluding implementation details).

Table 1. Partial contents of IIDB

Representation	Meaning	Ext	Sub
IETF	Internet Engineering Task Force	-	ietf
3978	RFC	-	doc
3978	OpenTTD game (masterserver and content service)	TCP;UDP	tcp
Jon Postel	RFC00001	img:lnk;text:url	human
protocol	rules determining the format and transmission of data	-	wordnet

Approaching the formal notations and definitions of WordNet [Miller 1995], the IIDB is defined as $W_I = (f, s, e, d)$ where f is a *form* composed by a string over a finite alphabet, s is *sense* got from a given set of meanings found in the unstructured bases (as RFC repository), e is an *extension* which complements s and d is the sub-domain to which *form* s belongs.

Hence, IIDB is a data set that covers the Internet Infrastructure domain and can be used for quick access not only by IEs but also by third parties and will help build the KB and support to update the KB and will be used as learning content for ML algorithms. The first three data bases which initially make up the IIDB – *iidb.rfc* (3,903 MBytes), *iidb.person* (4,961 MBytes) and *iidb.acronym* (9,590 MBytes) – are available in the repositories Open System Foundation (OSF) [Braga et al. 2019].

2.1. The value of IIDB

For several reasons it is necessary to find in which RFC an acronym has been defined. For example, a way to improve knowledge in a specific subject, which the acronym stands for. Having the number of the RFC in which it was defined and the RFCs that refer it, other bases can also be searched for the purpose of refining the knowledge. This set of research can contribute to become better the agents learning.

It is immediate to research by words in the acronym meaning. In this way, via the acronym it is possible to identify which RFCs treat the subject referenced by the word. For example when looking for 'NAT' we can get to RFC05720 and RFC06346, which should address the subject Network Address Translation, one of the meanings of the acronym (lines 5 and 6 in Table 4).

It is also possible to immediately identify which RFCs refer to an acronym or words of their meaning. Such facility are suitable for AS administrators, as well as technicians and researchers interested in interacting with RFCs.

The above privileges are appropriate, given the storage form of the IIDB bases, for testing, training and learning the ML algorithms.

Table 2 shows a set of meanings extracted from the corpus of the RFCs, showing that there are ambiguities in the meaning of the acronyms.

Table 2. Different meanings of RFC acronym

#	Acronym	Description	Document
1	RFC	REQUEST FOR COMMENTS	RFC01175
2	RFC	Request For Comment	RFC00199
3	RFC	Request For Comments	RFC00724
4	RFC	Request For Connection	RFC00033
5	RFC	Request-For-Connection	RFC00663
6	RFC	Requests-For-Connection	RFC00054
7	RFCs	REQUESTS FOR COMMENTS	RFC01175
8	RFCs	Request For Connections	RFC00671
9	RFCs	Requests For Comment	RFC08280
10	RFCs	Requests For Comments	RFC00661

Accented, included for humans are the ambiguities represented by lines 2 and 4. But for machines, lines 1 through 6 are ambiguous and so lines 7 to 10. Ambiguities are common in documents where there is no rigidity in the patterns of their composition. The bases of the IIDB allow disambiguation of the acronyms by references to the RFCs and, eventually, to their authors.

In the repository the IIDB is a Jupyter[3] notebook, with some Python scripts examples of using of the data bases.

3. Experimental design, material and methods

The techniques, features, and facilities used to construct the three bases that make up the IIDB are discussed below.

3.1. iibd.rfc and iibd.person

The *iibd.rfc* and *iibd.person* data bases were created based on data available in RFC Index web page[4]. For example, the information regarding the RFC8039 [Shpiner et al. 2016], is:

```
8039 Multipath Time Synchronization A. Shpiner, R. Tse, C.
Schelp, T. Mizrahi [ December 2016 ] (TXT = 39763) (Status:
EXPERIMENTAL) (Stream: IETF, Area: int, WG: tictoc) (DOI:
10.17487/RFC8039)
```

A computer program processed this information to fit the following pattern, arbitrarily defined:

```
'number': 8039 'title': 'Multipath Time Synchronization'
'author': 'first': 'A' 'second': '' 'last': 'Shpiner'
'author': 'first': 'R' 'second': '' 'last': 'Tse' 'author':
'first': 'C' 'second': '' 'last': 'Schelp' 'author':
'first': 'T' 'second': '' 'last': 'Mizrahi' 'nauthor': 4
'date': 'year': 2016 'month': December 'day': '' 'status':
'EXPERIMENTAL' 'stream': 'IETF'
```

[3]https://jupyter.org/index.html

[4]https://www.rfc-editor.org/rfc-index2.html

From the above pattern, other appropriate computer program captured the data to fill the two data bases (*iibd.rfc* and *iibd.person*), according to the respective templates presented in Listings 1 and 2.

Listing 1. iidb.rfc template

```
{
    "representation": representation,
    "meaning": {
        "title": title,
        "year": year,
        "month": month,
        "day": day,
        "status": status,
        "stream": stream,
        "words": {
            "volume": "len(text)",
            "vocabulary": "len(set(text))"
        },
        "lexical_diversity": "vocabulary/volume",
    },
    "extension": "",
    "subdomain": "rfc"
}
```

3.2. iibd.acronym

3.2.1. The construction process

Figure 2 shows the four steps that allow to prepare the RFCs for the extraction of acronyms.

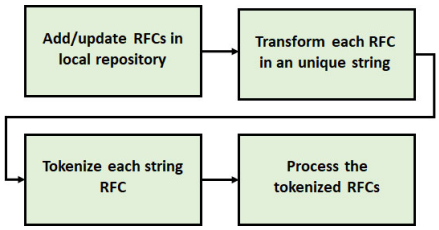


Figure 2. Preparing to process RFCs

Following the figure, the process of add/update RFCs to a local directory based on a search using the *Document Retrieval* features of the RFC-Editor[5].

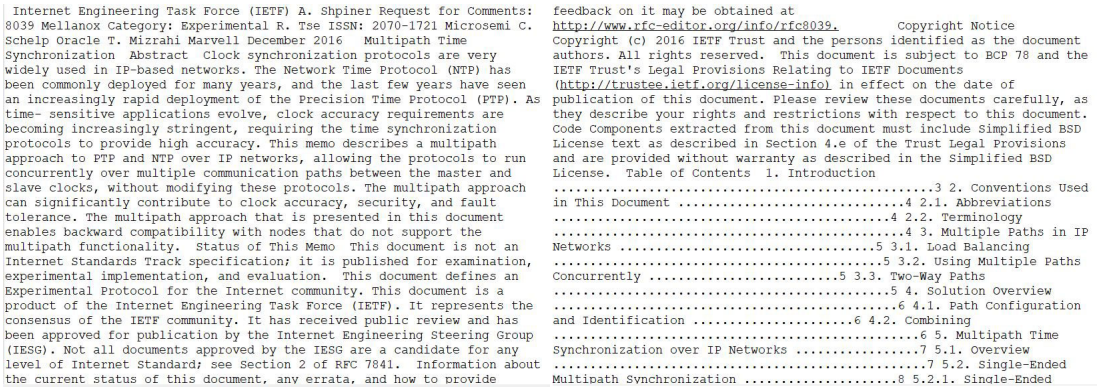
Listing 2. iidb.person template

```
{
    "representation": {
        "firstname": "",
```

[5]https://www.rfc-editor.org/retrieve/

```
        "secondname": "",
        "lastname": "",
    },
    "meaning": {
        "birthdate": "yyyymmdd",
        "deathdate": "yyyymmdd",
        "email": "",
        "gender": "",
        "company": "",
        "orcid": "",
        "photo": "",
        "authorship": [{
            "type_publication": type_publication,
            "id": number,
            "doi": doi,
            "author_seq": seq,
        }]
    },
    "extension": "",
    "subdomain": "human"
}
```

An RFC is also published in text format such as RFC8039[6] (an arbitrary choice). By transforming the RFC8039 and others one into a single string as can be seen in Figure 3 and then using the tokenize techniques, available in NLTK, we construct the corpus of the RFCs, that is, a bunch of properly formatted documents, gathered in a directory [Perkins 2014].



Internet Engineering Task Force (IETF) A. Shpiner Request for Comments: 8039 Mellanox Category: Experimental R. Tse ISSN: 2070-1721 Microsemi C. Schelp Oracle T. Mizrahi Marvell December 2016 Multipath Time Synchronization Abstract Clock synchronization protocols are very widely used in IP-based networks. The Network Time Protocol (NTP) has been commonly deployed for many years, and the last few years have seen an increasingly rapid deployment of the Precision Time Protocol (PTP). As time- sensitive applications evolve, clock accuracy requirements are becoming increasingly stringent, requiring the time synchronization protocols to provide high accuracy. This memo describes a multipath approach to PTP and NTP over IP networks, allowing the protocols to run concurrently over multiple communication paths between the master and slave clocks, without modifying these protocols. The multipath approach can significantly contribute to clock accuracy, security, and fault tolerance. The multipath approach that is presented in this document enables backward compatibility with nodes that do not support the multipath functionality. Status of This Memo This document is not an Internet Standards Track specification; it is published for examination, experimental implementation, and evaluation. This document defines an Experimental Protocol for the Internet community. This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 7841. Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc8039. Copyright Notice Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved. This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License. Table of Contents 1. Introduction .....3 2. Conventions Used in This Document .....4 2.1. Abbreviations .....4 2.2. Terminology .....4 3. Multiple Paths in IP Networks .....5 3.1. Load Balancing .....5 3.2. Using Multiple Paths Concurrently .....5 3.3. Two-Way Paths .....5 4. Solution Overview .....6 4.1. Path Configuration and Identification .....6 4.2. Combining .....6 5. Multipath Time Synchronization over IP Networks .....7 5.1. Overview .....7 5.2. Single-Ended Multipath Synchronization .....8 5.2.1. Single-Ended

Figure 3. Reduced RFC8039 text (partial view)

Thus, the corpus of RFCs facilitates the extraction of acronyms. This operation using an algorithm described below allows us to prepare some intermediate files that facilitate the availability of the acronyms data base as desired.

[6]https://tools.ietf.org/rfc/rfc8039.txt

3.2.2. Acronym in the context of RFCs

Acronyms and their expansions (or meaning), when first introduced in the text, are usually adjacent [Osiek et al. 2010]. In RFCs the universally used standard has the format *expansion (acronym)*, one of four formats identified by Pustejovsky et al. [Pustejovsky et al. 2001]. In this format, the (ACRONYM) will always be shown in capital letters. Usually, the expansion is composed of words with the first letters in upper case. Respecting this pattern and using some tools available in the Python language, and NLTK we extract the acronyms of the RFCs following the scheme shown in Figure 4.

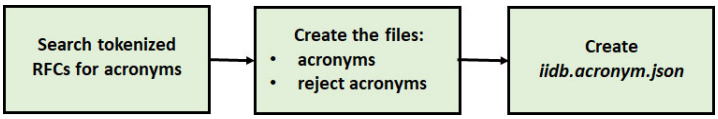


Figure 4. Preparing to process RFCs

On the other hand, the purpose of the *iibd.acronym* data base is to store acronyms and the respective RFC it was first occurred, in which RFCs it was referenced and how many times it appeared in each of them.

An acronym is a word created from the initial components of a phrase or name, called the expansion [Jacobs et al. 2018]. An acronym can be short-lived; if it was used in an RFC and is never referenced again. Also, an acronym can have more than one meaning. This is called *polysemy*. Polysemy, or lexical ambiguity, is the property of some words to have multiple meanings or senses [Moldovan and Novischi 2004]. In linguistics, *disambiguation*[7] refers to the process of explaining the message that has more than one meaning. To try to appease the disambiguation, the same acronym may appear several times in *iidb.acronym*, with divergent meanings, but referencing the RFC where it was first quoted and at other times.

The acronyms for this work were taken from all RFCs up to RFC08540 [Stewart et al. 2019]. The algorithm used to extract the acronyms from RFCs is summarized in Figure 5.

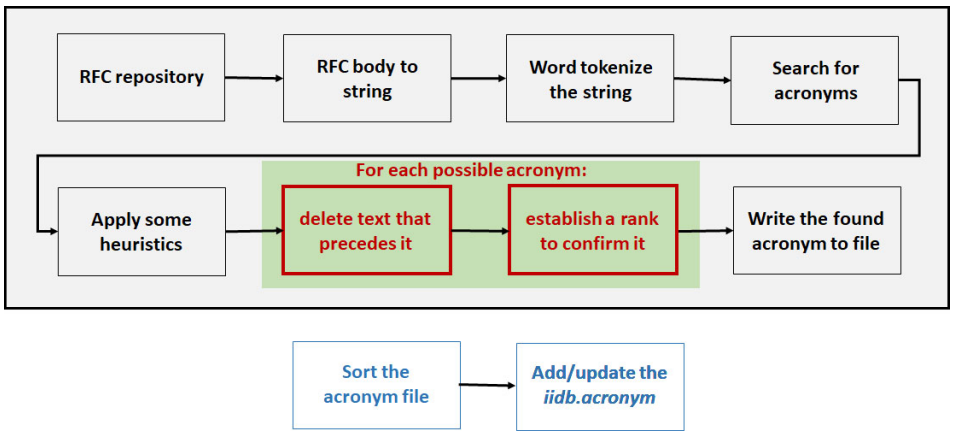


Figure 5. Algorithm used to extract acronyms from RFCs

[7]https://en.wikipedia.org/wiki/Word-sense_disambiguation

Three steps of the algorithm are relevant: *Apply some heuristics*, *delete text that precedes it* and *establish a rank to confirm it*. The last two steps are part of the kernel of the algorithm and considered to be primarily responsible for its success.

3.2.3. Step 1: "Apply some heuristics"

The requirements of this step can be summarized in the following topics:

- An acronym only exists if it is between '(' and ')'.
- The possible acronym must have more than 1 character.
- If the possible acronym is numeric, ignore it.
- If the possible acronym is all lowercase, ignore it.
- The words are stacked until a ')' is found. If an acronym is found, the stack is destroyed. So if there is not in the stack, at least the number of items corresponding to the number of letters found between '(' and ')', there is no acronym.
- It is acceptable that the acronym has the characters '&', '/' or '-'.
- It is acceptable '.' if it appears by following each of all the letters of the possible acronym. In this case all the letters should be uppercase.
- We make the assumption that an acronym's meaning (or expansion) lies before to the acronym. The reason for this assumption is associated with the pattern followed in the RFCs.
- Some authors consider that if the acronym contains numerical letters, its preceding letter (if there is any) or its following letter is repeated that many times to create a new acronym. This newly-created acronym is also used to find possible expansions by using other rules. For example, "W3C" can be changed to "WWWC" in order to discover W3C (World Wide Web Consortium) [Ji et al. 2008]. But, this does not work, in SS7 Application Part (S7AP), defined in RFC02719 [Ong et al. 1999]. The algorithm adopts the criterion of reproducing the letters on the left, when it finds 2 or 3 in the possible acronym.
- Some compound words such as 'multiprotocol' have been separated when are part of the acronym's meaning these words were given an additional weight, to strengthen the acronym. For example, Multiprotocol Label Switching (MPLS).

3.2.4. Step 2: "delete text that precedes it"

When the algorithm finds an acronym such as **(PPP)**, it pops up three previously stacked words. Probably, the three words unstacked will be: 'Protocol', 'Point-to-Point' and a third word whatsoever, say 'word3'. Arranged, the result would be: 'word3 Point-to-Point Protocol'. Thus, the algorithm in this step eliminates the word 'word3' concluding that the result of the meaning of 'PPP' will be: 'Point-to-Point Protocol. Once this is done, the next step should be to ensure that, indeed, this is the result.

3.2.5. Step 3: ”establish a rank to confirm it”

Keeping the **PPP** acronym example, this step removes the hyphens (’-’) and verifies that there is a *stop word*[8] (the ’to’) in the meaning of the acronym. Then the algorithm removes the *stop word*, adding to the acronym **PPP**, a negative weight, by having such a stop word. Then the algorithm continues analyzing the meaning and for each word that matches its first letter, with its position in the acronym, also receives a negative weight. When there is no coincidence, the weight is positive. In the example ’SS7 Application Part (S7AP)’, the acronym and its meaning are immediately accepted because their weight will be sufficiently negative to maintain the valid acronym ranking. Finally, let’s look at the case of the acronym **W3C**. This acronym stands for a *polysemy*. The following meanings were found: (1) ’World Wide Web Consortium’ (in the RFC05945), and (2) ’Worldwide Web Consortium’ (RFC02768). This acronym is well known and ambiguity may disappear in the context. This acronym is well known and ambiguity may disappear in this context. But the different RFCs, which refer to them, will certainly elucidate the issue. This is the case of the *polysemy* of the acronym **AIA**, with three meanings: (3) ’Association America’ (RFC04688), (4) ’Authority Info Access’ (RFC04809) and (5) ’Authority Information Access’ (RFC05280).

4. Conclusion

4.1. Application of the algorithm

The final result of the execution of this algorithm, implemented in Python, is represented by the numbers in Table 3.

Table 3. Statistics of the execution of the acronyms extraction algorithm

#	Representation	Meaning
1	RFCs processed (files)	$8,340$
2	Processing time (seconds)	$14,163.1$
3	Mean size of acronyms (chars)	3.36
4	Total number of acronyms extracted	$69,198$
5	Total number of acronyms extracted (no repetition)	$12,273$
6	Acronyms (no repetition) automatically confirmed	$11,098$
7	Precision	90.42%

The number of RFCs processed is less than the number of the last RFC processed, because there are numbers without RFCs, (1). The processing time is the result of the execution time given by the Sublime Text[9], (2). The acronyms average size, in number of characters, (3). The total number of acronyms extracted, with repetition, (4). Number of non-repeating acronyms from a non-human point of view, (6), representing 90.42% of (5), (7).

The file obtained in row (4) of Table 3 is sorted in alphabetical order and used to populate the *acronym.json* file, based on the template displayed in Listing 3.

[8] A word that is part of the meaning of the acronym, but usually is not represented in the acronym.

[9] https://www.sublimetext.com/

Listing 3. iidb.acronym template

```
{
        "representation": "",
        "meaning": {
                "acronym_of": "",
                "type": "",
                "appears-in": [{
                    "doc-id": "",
                    "times": ""
                }],
        "updated": [{
            "process": "automatic, agent, manual",
            "last_update": yyyymmdd
        }],
        },
        "extension": "",
        "subdomain": "acronym"
}
```

4.2. Acronym examples

Table 4 presents acronyms on which we will make some comments to reinforce the heuristic algorithm used.

Table 4. Acronym Examples

#	Name	Description	RFC
1	6LoWPANs	IPv6 over Low-Power Wireless Personal Area Networks	RFC06550
2	AAA	Authentication Authorization and Accounting	RFC05887
3	AAA	AUTHENTICATION AUTHORIZATION AND ACCOUNTING	RFC02881
4	AAA	Authentication Authorization and Accountability	RFC02888
5	CGN	Carrier-Grade NAT	RFC05720
6	CGNs	Carrier-Grade NATs	RFC06346
7	EBCDIC	Extended Binary-Coded Decimal Interchange Code	RFC00109
8	(FDV, also known as Jitter)	Frame Delay Variation	RFC07023
9	NETRJS	Remote Job Service	RFC00252
10	RJE	Remote Job Entry	RFC00105
11	RJE	Remote Job Entry Protocol	RFC00707
12	USASCII	USA Standard Code for Information Interchange	RFC00109
13	WRU	Who Are You	RFC00109

In **line 1**, '6LoWPANs'. If the number 6 appears in an acronym and does not follow a small 'v' letter, then it represents 'IPv6'. The same happens with the number 4, which turns into 'IPv4'. The acronyms in **lines 2–3** represent the same meaning (from the human point of view). The algorithm, however, considers them different and implements the two in the *iidb.acronym* file. It remains for the future work, the normalization of this difference. The acronym on **line 4**, has the seemingly strange meaning with the word 'Accountability'. This is a good example of the absence of standardization, whose proposal was put into future work. The RFC Editor should define between 'Accounting' and 'Accountability', for an acronym such as 'AAA'. **Lines 5–6** serve to illustrate the fact that a lowercase 's' at the end of the acronym does not influence the choice. This is the orientation that intelligent agents will receive when using IIDB. There is similarity to

this case with **lines 2–3**, relating to the acronym 'AAA'. The acronym of **line 7** has size 6. If you remove the hyphen from the second word of meaning, you have 6 words in the meaning. This is how the algorithm behaves. In **line 8**, very rarely is there an oversight of an author, escaping aggressively from the pattern of acronyms. The algorithm of this work fails in cases like this even though it is not complicated, in the context of a '(', end with a ',' instead of a ')'. But, this ran away from the more common pattern of acronyms in RFCs. In **line 9**, the algorithm will fail (but acknowledge and identify the failure), to *Remote Job Service* (NETRJS). In **lines 10–11**, same acronym with the word 'Protocol' in the end of the meaning. This word does not create ambiguity, because the algorithm admits the existence of it, or not at the end of the meaning. The acronym of **line 12**, 'USASCII' has size 7, but has one of the words of its meaning, integrally in the acronym: 'USA'. Then the size of the acronym becomes 5. However, it has a stop word, 'for' that reduces its size to 4. Acronym *Who Are You* (WRU), **line 13**, is only possible to be identified if we use a phonetic dictionary, since the characters **R** and **U** correspond to **ARE** and **YOU**. This was done in the algorithm, which gives each phoneme the appropriate punctuation.

4.3. Use of IIDB data bases

In the IIDB repository, as already mentioned, there is a file with some scripts in Python, using the data bases available there. By way of illustration, we show a simple script, but it uses two bases at the same time, to determine the first 30 most participant authors of RFCs. Figure 6 shows the program and on the right side, the result.

```python
import json

dir_json = "C:/Users/User/Google Drive/dev/wordIETF-1.1/json/"
#
# Read iidb.person
#
with open(dir_json+'iidb.person20190303.json', 'r') as fin:
    person = json.load(fin)

#
# Read iidb.rfc
#
with open(dir_json+'iidb.rfc20190303.json', 'r') as fin:
    rfcs = json.load(fin)
# Output the first 30 authors of RFCs by number of participation
dicta = {}
for i in range(len(person)):
    author = person[i]['representation']['firstname']+' '+
        person[i]['representation']['lastname']
    autoria = len(person[i]['meaning']['authorship'])
    dicta.update({author:autoria})
s = sorted(dicta.items(), key=lambda x: x[1], reverse=True)
for i in range(30):
    print(str(i+1)+':', s[i])
```

```
1: ('K. McCloghrie', 93)
2: ('H. Schulzrinne', 89)
3: ('D. Eastlake 3rd', 87)
4: ('R. Housley', 87)
5: ('Y. Rekhter', 83)
6: ('H. Tschofenig', 82)
7: ('J. Rosenberg', 70)
8: ('G. Camarillo', 69)
9: ('P. Hoffman', 68)
10: ('A. Farrel', 66)
11: ('D. Crocker', 63)
12: ('F. Baker', 61)
13: ('A. Melnikov', 60)
14: ('No Author', 59)
15: ('C. Perkins', 58)
16: ('J. Klensin', 56)
17: ('M. Rose', 55)
18: ('E. Rosen', 53)
19: ('D. Thaler', 53)
20: ('B. Aboba', 53)
21: ('B. Carpenter', 51)
22: ('G. Zorn', 51)
23: ('C. Pignataro', 51)
24: ('N. Freed', 49)
25: ('S. Turner', 46)
26: ('M. Boucadair', 46)
27: ('S. Bradner', 45)
28: ('J. Arkko', 45)
29: ('T. Nadeau', 45)
30: ('V. Cerf', 44)
```

Figure 6. First 30 most participant authors of RFCs.

5. Related Works

The importance of extracting acronyms is acclaimed by Sánchez and Isern [Sánchez and Isern 2011]: *The discovery of the definitions associated to an acronym is an important matter in order to support language processing and knowledge-related tasks as information retrieval, ontology mapping or question answering*. Xiaonan Ji and colleagues [Ji et al. 2008], consider that *techniques for being able to automatically identify acronym patterns are very important for enhancing a multitude of applications that rely upon search* and present a new approach to discover acronyms patterns.

Other techniques such as logical-algebraic equations that combine grammatical and semantic characteristics of words of substantive, attributive and verbal collocations types exploit WordNet resources [Khairova et al. 2018].

Unsupervised learning techniques have been and are used in the medical field, where ambiguity of acronyms occur mainly when the context is not delimited. If we see the acronym RA in a cardiology report, then it can be normalized to "right atrial"; otherwise, if it occurs in the context of a rheumatology note, it is likely to mean "rheumatoid arthritis" or "rheumatic arthritis" [Pakhomov 2002]. So the method of using the global context to solve the ambiguity of an acronym as it is done in the medical field is not an adequate solution to the ambiguity of the acronyms found in the RFCs. It is noted that ambiguity is often produced, by lexical error or typos of the authors, both very common, as can be seen. Also, other techniques as machine-learning-based approach to automatically build an acronym dictionary from texts are proposed [Jacobs et al. 2018].

A simple formula, where a $score$ is determined by Equation 1, and if it is below some *threshold*, then the pair is accepted. This was proposed by [Pustejovsky et al. 2001].

$$score = \frac{\#\ \ of\ \ \ words\ \ \ in\ \ \ the\ \ \ match}{\#\ \ of\ \ \ characters\ \ \ in\ \ \ the\ \ \ acronym} \tag{1}$$

Reinforcement learning techniques like *Markov chains* [Paulino et al. 2018] and unsupervised learning techniques like *Hidden Markov models* (HMMs) are a powerful probabilistic tools for modeling time series data, and have been applied with success to many language-related tasks such as part of speech tagging, speech recognition, text segmentation and topic detection [Freitag and McCallum 1999]. Several authors have used acronyms extraction in medical unstructured texts [Osiek et al. 2010] [Conroy and O'leary 2001]. Others one, in speech recognition [Rabiner 1989].

The issue associated specifically with the extraction of acronyms is intensely addressed in Manuel Zahariev's doctoral thesis [Zahariev 2004].

6. Future Works

RFCs are not ready to be evaluated by machines or smart codes. For example, *Monkey in the Middle* (MITM) should come closer to the machine understanding perspective being represented as (MitM), ensuring that *stop words* are in lower case, in the acronym. The absence of standards when writing an RFC creates ambiguity between acronyms, more often than was expected (CGN e CGNs, both with the same meaning). The IETF, as a careful standards-setting institution, should take better care of the standardization of its main document, since it is from RFCs that we can gain knowledge to provide intelligent agents. The authors believe and in the near future should lead to a draft proposing writing standards that could be required from the authors to facilitate the use of machine learning and other techniques of artificial intelligence. This kind of care should be extended to IETF peripheral institutions, such as *Regional Internet Registers* (RIRs), which produce a lot of important information for ASes administrators.

The algorithm used to extract the acronyms from RFCs can be improved to use machine learning techniques on rejected acronyms. Additionally, refinement of the files generated by the current algorithm could increase the accuracy of the result. Both al-

ternatives allow to produce appropriate results for measurements and evaluations of the acronyms capture algorithm [Batista et al. 2004]. This was not done in this project, because it was outside the goal of the global experiment.

Acknowledgment

This work is supported by CAPES – Brazilian Federal Agency for Support and Evaluation of Graduate Education within the Brazil's Ministry of Education, by national funds through FCT with reference UID/CEC/50021/2019 and by MackPesquisa. .

References

Batista, G. E. A. P. A., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, 6(1):20–29.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python*. " O'Reilly Media, Inc.".

Braga, J., Silva, J. N., Endo, P. T., and Omar, N. (2019). Autonomous Architecture Over Restricted Domains (A2RD). DOI 10.17605/OSF.IO/TKA9U. Available at https://osf.io/tka9u/. Acessed: 19 Mar 2019.

Braga, J., Silva, J. N., Endo, P. T., Ribas, J., and Omar, N. (2018). Blockchain to Improve Security, Knowledge and Collaboration Inter-Agent Communication over Restrict Domains of the Internet Infrastructure. In *Proceeding of CSBC 2018 - V Workshop pre IETF*, pages 61–73, Natal, RN Brazil.

Colel, R., Callon, R., Gardner, E., and Rekhter, Y. (May 1994). Guidelines for OSI NSAP Allocation in the Internet . Technical report, RFC Editor. RFC1629.

Conroy, J. M. and O'leary, D. P. (2001). Text summarization via hidden Markov models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 406–407.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Freitag, D. and McCallum, A. (1999). Information extraction with hmms and shrinkage. In *Proceedings of the AAAI-99 workshop on machine learning for information extraction*, pages 31–36. Orlando, Florida.

Hares, S. and Katz, D. (December 1989). Administrative Domains and Routing Domains: A model for routing in the Internet. Technical report, RFC Editor. RFC113.

Isotani, S. and Bittencourt, I. I. (2015). *Dados abertos conectados*. Novatec Editora, São Paulo, SP, Brasil.

Jacobs, K., Itai, A., and Wintner, S. (2018). Acronyms: identification, expansion and disambiguation. *Annals of Mathematics and Artificial Intelligence*, pages 1–16.

Ji, X., Xu, G., Bailey, J., and Li, H. (2008). Mining, ranking, and using acronym patterns. In *Asia-Pacific Web Conference*, pages 371–382. Springer.

Khairova, N., Petrasova, S., Lewoniewski, W., Mamyrbayev, O., and Mukhsina, K. (2018). Automatic extraction of synonymous collocation pairs from a text corpus. In *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 485–488. IEEE.

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

Moldovan, D. and Novischi, A. (2004). Word sense disambiguation of wordnet glosses. *Computer Speech & Language*, 18(3):301–317.

Musumeci, F., Rottondi, C., Nag, A., Macaluso, I., Zibar, D., Ruffini, M., and Tornatore, M. (2018). A Survey on Application of Machine Learning Techniques in Optical Networks. *IEEE Communications Surveys & Tutorials*, pages 1–1.

Ong, L., Rytina, I., Garcia, M., Schwarzbauer, H., Coene, L., Lin, H., Juhasz, I., Holdrege, M., and Sharp, C. (October 1999). Framework Architecture for Signaling Transport. Technical report, RFC Editor. RFC2719.

Osiek, B. A., Xexéo, G., and de Carvalho, L. A. V. (2010). A language-independent acronym extraction from biomedical texts with hidden markov models. *IEEE Transactions on Biomedical Engineering*, 57(11):2677–2688.

Pakhomov, S. (2002). Semi-supervised maximum entropy based approach to acronym and abbreviation normalization in medical texts. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.

Paulino, C. D. M., Turkman, M. A. A., and Murteira, B. (2018). *Estatística Bayesiana*. Fundação Calouste Gulbenkian, second edition.

Perkins, J. (2014). *Python 3 text processing with NLTK 3 cookbook*. Packt Publishing Ltd.

Poole, D. L. and Mackworth, A. K. (2010). *Artificial Intelligence: foundations of computational agents*. Cambridge University Press.

Pustejovsky, J., Castano, J., Cochran, B., Kotecki, M., and Morrell, M. (2001). Automatic extraction of acronym-meaning pairs from medline databases. *Studies in health technology and informatics*, 84(1):371–375.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Sánchez, D. and Isern, D. (2011). Automatic extraction of acronym definitions from theWeb. *Applied Intelligence*, 34(2):311–327.

Shpiner, A., Tse, R., Schelp, C., and Mizrahi, T. (December 2016). Multipath Time Synchronization. Technical report, RFC Editor. RFC8039.

Stewart, R., Tuexen, M., and Proshin, M. (February 2019). Stream Control Transmission Protocol: Errata and Issues in RFC 4960. Technical report, RFC Editor. RFC8540.

Zahariev, M. (2004). *A(Acronyms)*. PhD thesis, Simon Fraser University.