A simple method to extract abbreviations within a document using regular expressions

Christian Sánchez, Paloma Martínez

Computer Science Department, Universidad Carlos III of Madrid Avd. Universidad, 30, Leganés, 28911, Madrid, Spain

Abstract. Biomedical Abbreviation Recognition and Resolution (BARR) is an evaluation track of the 2nd Human Language Technologies for Iberian languages (IberEval) workshop, a workshop series organized by the Spanish Natural Language Processing Society (SEPLN). In this first edition of BARR, we focus on the discovery of biomedical entities and abbreviation, and relating detected abbreviations with their long forms. This paper describes the approach and the system presented in the sub-track 2, which consists in offers a method to extract abbreviations within a document using regular expressions.

1 Introduction

Many clinical documents are created in a daily basis, most of them contain abbreviations for common medical and clinical terms, names of diseases, symptoms, etc., the correct interpretation of them could be sometimes confusing for patients and even for medical professionals. This also adds some workload, because find, retrieve and interpret an abbreviation could often includes not just analyse the term but the whole document context.

There is some research that proposes certain solutions and approaches for the problem, but most of it is focused on analysing text written in English, in this context the BARR2 track has the aim to promote the development and evaluation of clinical abbreviation identification systems by providing Gold Standard training and test corpora manually annotated by domain experts with abbreviation-definition pairs within abstracts of clinical texts and clinical case studies written in Spanish.

Our participation was focused on the sub-track 2: ***provide resolution of short forms regardless whether its definition is mentioned within the actual document***. For this approach, and in line with our participation in the previous BARR track, we refer to an abbreviation as a *Short Form (SF)* and the definition as the *Long Form (LF)*.

This paper is organized as follows: Section 2 describes our proposed approach. Section 3 presents evaluation and results. Finally, conclusions and future work are discussed in Section 4.

2 Proposed Approach

The main goal was to propose a solution for the sub-track 2. This proposal was based on our previous work *A proposed system to identify and extract abbreviation definitions in*

Spanish biomedical texts for the Biomedical Abbreviation Recognition and Resolution (BARR) 2017 [4]. To accomplish this, first, we assumed that an abbreviation or *short form* could appear many times in the document, its length should be between 2 and 8 characters, and have just one and single definition. Secondly, we use an external source to obtain the definition, so we declined to evaluate the content of the document, i.e.:

A la exploración física se observaba paraparesia con amioatrofia por desuso de EEII

In this example we consider ***EEII*** as a *short form* or abbreviation, and the actual definition or *long form* is not provided within the text.

Using the mentioned assumptions as guidelines, we divided the system process into the following tasks:

2.1 Prepare and organize the definitions

We used the *Diccionario de Siglas Médicas* [1] as the main source for the definitions. All the terms were extracted from there, stored in a database and exposed as a service in a REST API. The total number of terms contained in the dictionary and exported to the database was 3386. This service was meant to be used as part of the system used in the presented approach.

Definitions are returned as a list of *key-value* objects, composed by the *short form* and the *long form* or definition, i.e. for the abbreviation *EEII*:

```
[
    {
        "long_form": "Extremidades inferiores",
        "short_form": "EEII"
    },
    {
        "long_form": "Extremidades izquierdas",
        "short_form": "EEII"
    }
]
```

Some abbreviations could have more than one definition, in this way it is possible to obtain all the known definitions for a given abbreviation.

2.2 Detect Short Forms

For this task we use a *Perl 6* script which parses all the documents one by one to obtain all the *short forms* found in the text. Short form identification was performed using regular expressions[1] . The set of rules was based in our previous work, but some improvements were added. A total of 5 regular expressions were used for the proposed system.

One of the improvements in one of the regular expression used was the following;

[1] https://docs.perl6.org/language/regexes

```
<[a..z]>? \-? <[A..Z]> ** {1..8} <[a..z\_\-\/]>? <[A..Z0..9]>+
    <[a..z]>*
```

This regular expression matches (from left to right):

- Zero or one lowercase letter
- Zero or one "-" character
- Between 1 and 8 uppercase letters
- Zero or one lowercase letter or "_", "-", "\" characters
- One or more uppercase letter or number
- Zero or more lowercase letters

Also, to match abbreviations in the form *gr/dl*, the following regexp was used:

```
<[a..zA..Z]>**{1..4} \/ <[\w]>**{1..4}
```

This regular expression matches (from left to right):

- Between 1 and 4 lowercase or uppercase letters
- The character /
- Between 1 and 4 word characters (a-z, A-Z, 0-9, including the character _)

The whole document is parsed and all the matches found are stored in a list. This detection also stores the position of the matched *short form* in the text. Once the document is processed, the next step is obtain the definition of each of them.

2.3 Get Long Forms

Using the REST API provided for the definitions database, it was possible for the script to make GET requests, via HTTP, for each of the abbreviations found in the document. If a definition was not found in the database, the script discarded the current abbreviation processed and continued with the next match.

This step was executed every time the script needed to find a definition, if a definition provided was associated with an abbreviation, the script marked it as done and did not execute this step even if there were more matches in the document, this provided a better performance for the system.

2.4 Process Long Forms

When a response was provided by the API, the script continued with the next step, which was to process and to obtain the supposedly right definition for the abbreviation. In this step there were two possibilities:

If the response contained just one definition, the script used it and marked it as the definitive for the current evaluated abbreviation and started the task with the next item in the list.

If the response contained two or more definitions, the script performed another set of actions for each result:

- Normalize part of the text, get the content until the start offset of abbreviation. This normalization includes remove stop words and stemming the text using the *Perl 6* modules ***Lingua::Stopwords*** [2] and ***Lingua::Stem::Es*** [3].
- Normalize text of definition, remove stop words and stemming the text using the same tools as the previous step.
- Extract the same amount of words from the normalized definition as the length of characters from the abbreviation, beginning from the start offset of the abbreviation and moving to the left side, so if the abbreviation is *EEII* this step should get four (4) stemmed words from the normalized text.
- Perform and intersection operation to get a list that contains only the elements common to both the normalized text and the normalized definition, and return the total elements found.

In the case that not matches were found, the steps above were repeated, but instead of extracting a number of words from the normalized text, the intersection operation was made with all the text until the start offset of the abbreviation. That way a wide range of stemmed words were compared which provided a better context and more opportunities to find similarities in both texts.

Once all the *long forms* were processed, the script selected the one which more intersected elements and use it as the definition. As a final step, the script obtained the lemmatized version of the definition using the *Python* library ***pattern*** [4]

3 Evaluation and Results

For this sub-task at BARR2 the primary evaluation metric used consisted in precision, recall, and f-score of the predictions against manual gold standard[3]. A corpus consisting in a manually labeled collection of Spanish medical abstracts constructed using a customized version of AnnotateIt, BRAT as well as using the Markyt annotation system [9] was released for the organizers to test the systems[2].

The results for our first test with the training data (a total of 4260 annotations provided) were:

```
PRECISION = 0.5130085 = 1459.5092 / 2845
RECALL = 0.3426078 = 1459.5092 / 4260
F–MEASURE = 0.41084
```

After some adjustment in the rules for *short form* detection and cleanup some texts in the definitions stored in the database we got an improvement in the results:

```
PRECISION = 0.722437 = 1526.5094 / 2113
RECALL = 0.35833555 = 1526.5094 / 4260
F–MEASURE = 0.47905517
```

[2] http://modules.perl6.org/dist/Lingua::Stopwords:cpan:CHSANCH
[3] http://modules.perl6.org/dist/Lingua::Stem::Es:cpan:CHSANCH
[4] https://www.clips.uantwerpen.be/pages/pattern-es

The evaluation scores obtained for our 3 submitted predictions were:

```
Precision: 74.93
Recall: 37.69
F1: 50.16
```

There were some issues that the organizers noticed in this sub-track: definitions could appear in different forms, there are variants of some of the definitions, and some typos; all of them could affect the results on some level.

4 Conclusions and future work

For this sub-task we relayed just in one dictionary, which provides a good resource for definitions in the medical field, more sources are needed to improve the results. This proposal offers a solution in this specific field, but it could be extended to analyze documents related to other fields.

Another interesting improvement could be to add some Machine Learning processes to classify texts and provided an accurate selection of the definition of an abbreviation in the context of the document processed.

There were many missed definitions. An attempt to get and stored definitions for missed abbreviations matches using externals sources could be an important improvement. Finally in addition, apply some methods to identify and extract definitions within the document processed, which was the main goal of the sub-track 1.

References

1. Diccionario de Siglas Médicas. Ministerio de Sanidad y Consumo (2016)
2. Intxaurrondo, A., Marimon, M., Gonzalez-Agirre, A., Lopez-Martin, J., Betanco, H.R., Santamaría, J., Villegas, M., Krallinger, M.: Finding mentions of abbreviations and their definitions in spanish clinical cases: the barr2 shared task evaluation results. SEPLN (2018)
3. Intxaurrondo, A., de la Torre, J., Betanco, H.R., andJ A. Lopez-Martin, M.M., Gonzalez-Agirre, A., Santamaría, J., Villegas, M., Krallinger, M.: Resources, guidelines and annotations for the recognition, definition resolution and concept normalization of spanish clinical abbreviations: the barr2 corpus. SEPLN (2018)
4. Sánchez, C., Martínez, P.: A proposed system to identify and extract abbreviation definitions in spanish biomedical texts for the biomedical abbreviation recognition and resolution (barr) 2017. BARR IBEREVAL (2017)