A Machine Learning Approach to Clinical Terms Normalization

José Castaño[1], Hernán Berinsky[2], Hee Park[2], David Pérez[2], Pilar Ávila[2],
Laura Gambarte[2], Sonia Benítez[2], Daniel Luna[2], Fernando Campos[2] and Sofía Zanetti[2]

[1]Depto. Computación, FCEyN, Universidad de Buenos Aires
jcastano@dc.uba.ar

[2]Departamento de Informática en Salud, Hospital Italiano de Buenos Aires
{firstname.lastname}@hospitalitaliano.org.ar

Abstract

We propose a machine learning approach for semantic recognition and normalization of clinical term descriptions. Clinical terms considered here are noisy descriptions in Spanish language written by health care professionals in our electronic health record system. These description terms contain clinical findings, family history, suspected disease, among other categories of concepts. Descriptions are usually very short texts presenting high lexical variability containing synonymy, acronyms, abbreviations and typographical errors. Mapping description terms to normalized descriptions requires medical expertise which makes it difficult to develop a rule-based knowledge engineering approach. In order to build a training dataset we use those descriptions that have been previously matched by terminologists to the hospital thesaurus database. We generate a set of feature vectors based on pairs of descriptions involving their individual and joint characteristics. We propose an unsupervised learning approach to discover term equivalence classes including synonyms, abbreviations, acronyms and frequent typographical errors. We evaluate different combinations of features to train MaxEnt and XGBoost models. Our system achieves an F_1 score of 89% on the Hospital Italiano de Buenos Aires (HIBA) problem list.

1 Introduction

Some electronic health records (EHR) implementations allow users to introduce free text descriptions to capture clinical problems information enabling higher level of expressiveness and flexibility to physicians. Those descriptions must be encoded according to their meaning in order to allow the information to be consumed by other systems. Descriptions are grouped into concepts according to the meaning. The following descriptions correspond to the same concept[1]:

(1) neoplasia maligna de pulmón
neoplasia malign of lung
'Malignant tumor of lung'

(2) cáncer pulmonar
cancer lung-of
'lung cancer'

(3) ca pulmonar
ca lung-of
'lung cancer'

(4) cáncer de pulmón desde 2009
cancer of lung since 2009
'cancer of the lung since 2009'

Free text descriptions written by health care professionals contain typos as in *cancer plumnoar*: a variation of description (2). It should be noted also that description (4) does not represent a synonym in a strict terminological sense. However it represents the same concept because the string *desde 2009* (since 2009) does not add relevant information from a *problem list* perspective (in the sense of EHR and terminology tradition (Van Vleck et al., 2008)).

Mapping strings to concepts has been a long standing problem in BioNLP, string similarity techniques as well as machine learning approaches have been applied. Automatic mapping of key concepts from text in clinical notes to a reference terminology is an important task to achieve, in order to extract clinical information present in notes and patient reports. One of the problems of bio-

[1] In these examples the Spanish description is followed by the word-for-word English gloss and then the English translation.

Proceedings of the 15th Workshop on Biomedical Natural Language Processing, pages 1–11,
Berlin, Germany, August 12, 2016. ©2016 Association for Computational Linguistics

medical data integration is variation of terms usage. Exact string matching often fails to associate a string with its bio-medical concept (represented by an ID or accession number in the database) due to differences of string occurrences. Soft string matching algorithms are able to find the relevant concept by considering the string similarity between candidate strings. However, the accuracy of soft matching highly depends on the similarity measure employed. String similarity techniques have been applied to a variety of problems in BioNLP, such as UMLS concepts normalization (Aronson and Lang, 2010; Wellner et al., 2005; Rudniy et al., 2014), UMLS clinical terms (Kate, 2016), disease normalization (Leaman et al., 2013; Kang et al., 2013), gene and protein names (Kim and Park, 2004; Tsuruoka et al., 2007; Tsuruoka et al., 2008; Fang et al., 2006; Wermter et al., 2009), to interface terminologies (Rosenbloom et al., 2006) and different databases (Sun, 2004).

String similarity can be used for named entity recognition (SNOMED-CT taggers) and reference resolution (Castaño et al., 2002; Lin and Liang, 2004; C. et al., 2003) alias extraction (Yu and Agichtein, 2003), acronym-expansion extraction, e.g. (Pustejovsky et al., 2001).

On a similar view there are a number of works on automated clinical coding (Friedman et al., 2004; Pakhomov et al., 2006; Patrick et al., 2006; Suominen et al., 2008; Stanfill et al., 2010; Perotte et al., 2014).

This work explores traditional soft string matching methods along with n-gram character and word features in a machine learning approach using MaxEnt and XGBoost classifiers. An unsupervised learning approach to generate new features by detecting synonyms, abbreviations and typos is presented to improve classification performance. The models are compared to a baseline obtained by a vector space model configuration based on character n-grams and a TF-IDF weighting scheme, implemented in Apache Lucene.

The remainder of this paper is structured as follows: in Section 2, we describe the data set we used. In Section 3, we discuss the similarity metrics and similarity features for machine learning algorithms. In Section 4 we discuss the experimentation and results. Finally in Section 5 we report our conclusions and expected future work.

2 Description Terms Data-set

We build a data-set based on the problem list of Hospital Italiano de Buenos Aires (HIBA) interface terminology (Lopez Osornio et al., 2007; Gambarte et al., 2007) which includes adequate synonym coverage and it is linked to the HIBA thesaurus. This thesaurus is built upon the Spanish version of SNOMED-CT, while extending it with new concepts and additional synonym terms.

Following SNOMED-CT and other thesauri, terms in the thesaurus are grouped by concepts. The following terms are associated to the same concept.

(5) tabaquismo
smoking

(6) abuso de tabaco
tobacco consumption

We selected those clinical concepts that had at least 10 terms and no more than 100 for a given concept.[2] The set is composed of 151,513 terms and 5,222 concepts. The set of descriptions (D) was split in a training set (T) 70%, and an evaluation test set (E) 30%.

Descriptions in T were used to build a new dataset T_1 consisting of pairs of descriptions samples of the form $(d_1, d_2, value)$. Positive and negative samples were constructed in the following way:

- For each pair of descriptions $d_i, d_j \in T$ with $i \neq j$ such that d_i and d_j are associated to the same concept, we create a sample $(d_i, d_j, 1)$

- We split the set of descriptions T in corpus and query sets. We indexed with Apache Lucene the corpus set of descriptions using TF-IDF weights on n-gram characters. Using a description d as a query, a set of relevant and non-relevant results are retrieved. Relevant results are those descriptions d_i already stored as samples of pair of terms describing the same concept: (d, d_i, 1). Non-relevant results are those results d_j for which there is not a sample (d, d_j, 1) and therefore a sample $(d, d_j, 0)$ is created.

The training data-set (T_1) has 1,173,617 instances with 777,585 negative and 396,032 positive samples.

[2] Those concepts that had more than 100 terms were noisy, and were not considered relevant.

3 Methods for Computing Term Normalization

String similarity methods can be either character-based or token-based. Character-based approaches typically consist of variations of the edit-distance metric, like Levenshtein distance or longest common subsequence. Token-based approaches include the Jaccard similarity metric and the cosine similarity based on TF-IDF weighting schema. There are also hybrid token and character-based approaches. Soft-TFIDF (Cohen et al., 2003) includes not only exact matches but also close matches, using a threshold. Another approach uses n-grams of the target strings instead of the tokens (Cohen et al., 2003; Moreau et al., 2008; Köpcke and Rahm, 2010).

Many works have also focused on automatic methods for combining these string similarity measures using machine learning (Cohen and Richman, 2001; Belenko and Mooney, 2003; Wellner et al., 2005; Moreau et al., 2008).

In this section we explore a hybrid soft-TFIDF approach based on an n-gram character vector space model as well as other character-based and token-based similarity metrics. Next, we mention some limitations of combining the previous metrics due to information redundancy and lack of semantic information which produces false positive and false negative instances. We propose an usupervised machine learning approach which allows to capture semantic information.

3.1 Information retrieval and TF-IDF

We use an information retrieval (*IR*) Soft-TFIDF approach (Cohen et al., 2003) to match a new description to those terms already existing in the hospital thesaurus database. First, the set of known terms in the thesaurus are indexed with Lucene, where the collection of terms is represented in a Vector Space Model (VSM) using TF-IDF weights based on character n-grams. A new description is used as a query and the set of ranked descriptions terms with the corresponding scores is retrieved, being the highest ranked description the candidate term to associate the query with. The cosine similarity measure is used to obtain similarity scores.

However this approach will outcome both false positive and false negative results such as:

(7) sospecha de laringitis alérgica (query)
suspected (of) alergic laringytis

(8) sospecha de faringitis alérgica (false positive)
suspected (of) alergic pharyngitis

Due to the high string similarity score between *sospecha de laringitis alérgica* and *sospecha de faringitis alérgica* if either of them is not indexed as a concept, then the returned result is considered a match and therefore a false positive instance is obtained.

(9) neoplasia maligna de pulmón (query)
malignant tumor of lung

(10) cáncer pulmonar (false negative, not retrieved)
lung cancer

A low similarity score between *neoplasia maligna de pulmón* and *cáncer pulmonar* implies that the target string is not retrieved (i.e. it is not ranked above the threshold). Since the concept is just represented by *cáncer pulmonar*, the string *neoplasia maligna de pulmón* is a false negative instance.

Figure 1 shows overlapping distribution of scores. The positive match curve represents the score (cosine similarity) distribution of query and retrieved string pairs that represent the same concept. It shows higher average score than negative match. As threshold score increases, false negative cases increase and false positive cases decrease.

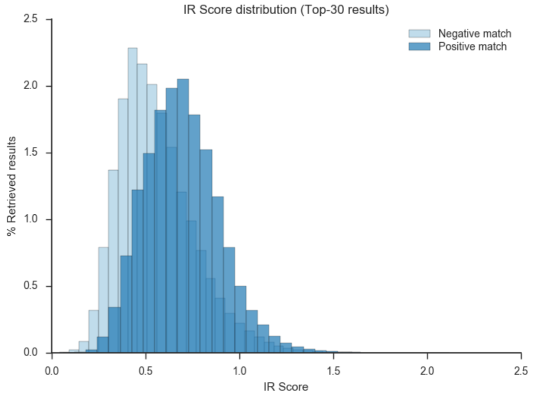


Figure 1: IR Score distribution (normalized histograms)

Given a query, it is not known whether relevant information exists or not in the indexed dataset, and the term with the highest score is not necessarily a desired result. The performance of matching the query with the highest ranked term can be measured using *precision*, *recall* and F_1 metrics.

In a Soft-TFIDF approach is possible to control precision/recall trade-off considering a threshold t as shown in Figure 2. The algorithm returns the highest ranked term if $score \geq t$. Higher values of t increase precision but recall is decreased.

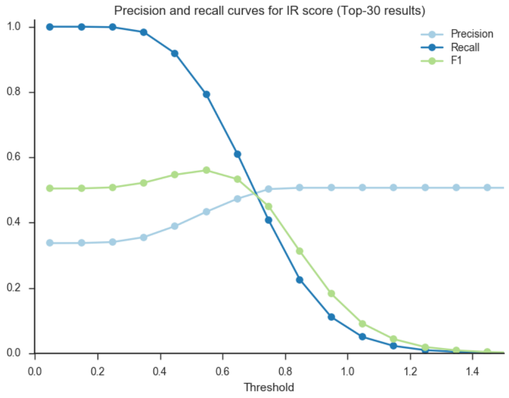


Figure 2: Precision, recall and F1 measure
Precision(t), recall(t) and $F_1(t)$ show measures for the results returned by the IR system where $score \geq t$ (when increasing t, precision(t) increases and recall(t) decreases)

3.2 String similarity metrics

String similarity metrics have been used combined with *IR* TF-IDF approaches. Using traditional string similarity and distance metrics like *Damerau-Levenshtein* or *Longest common subsequence* allow to increase to some extent precision. It is possible to score results using a string metric or combine together with *IR* scores using some rules or formulas and thresholds. Using a set of scores it is possible to use machine learning models to classify relevant/non-relevant result.

Even though there are other string metric measures that can be combined, some of them are very much related. For example Damerau-Levenshtein ($DamLev_{dist}$) distance allows an additional edit operation respect to Levenshtein (Lev_{dist}) distance, then $DamLev_{dist} \leq Lev_{dist}$. Also Jaccard and Sorensen-Dice similarity metrics present a high correlation. In Table 1 we show pairwise correlations between Damerau-Levenshtein ratio, Longest common sub-sequence, Sorensen-Dice, Jaro-Winkler and Jaccard coefficient metrics.

Due to high correlation between Jaccard and Sorensen-Dice, we can choose one of them, and in the same way with Damerau-Levenshtein and Longest common subsequence to fit a classification model using machine learning approach.

Metric	DamLev	LCS	SorDic	JarWin	Jac
DamLev	1.00	0.96	0.78	0.73	0.77
LCS		1.00	0.84	0.73	0.82
SorDic			1.00	0.65	0.99
JarWin				1.00	0.64
Jac					1.00

Table 1: Correlation (Pearson) between string metrics
Damerau-Levenshtein ratio (DamLev), Longest common subsequence (LCS), Sorensen-Dice (SorDic), Jaro-Winkler (JarWin) and Jaccard (Jac). Damerau-Levenshtein ratio is a transformation of Damerau-Levenshtein distance d using the formula M is the maximum lenght of s_1 and s_2.

By computing the principal components, the eigenvalues show that using the first k components the cumulative variance explained is 76% ($k = 1$), 93% ($k = 2$), 98% ($k = 3$), 99% ($k = 4$). This means that k new variables (linear combination of original metrics) explain those proportions of the total variance and we can also reduce redundant information.

Limitations to this approach are present both in false positive and false negative cases. It is a quantitative improvement but cases like those presented in examples (7-10) above, require a more sophisticated approach. Such approach must consider which modifications in a clinical term changes its meaning.

3.3 A machine learning approach to string matching

As it has been already observed in many other works, abbreviations, acronyms, synonyms and typos are sources of variation that generate terms with the same meaning. Table shows some examples from the Spanish dataset:[3]

Many pairs of description terms are very similar but they have different meaning as described by the following non-synonym pairs where only a character difference in a long string entails a different meaning:

(11) a. *sospecha de* **l***aringitis alérgica*
b. *sospecha de* **f***aringitis alérgica*

(12) a. *duelo por fallecimiento de* **m***adre*
b. *duelo por fallecimiento de* **p***adre*

(13) a. *sospecha de hi* **po***tiroidismo*
b. *sospecha de hi* **per***tiroidismo*

(14) a. *artr* **i***tis de tobillo*
b. *artr* **o***sis de tobillo*

[3] We do not include the translations from now on because the relevant information is the string similarity.

Alternative forms	*'meaning'*
sd, *sme*, *sind*, *sindr*	*sindrome*
izq, *izqdo*, *iz*	*izquierdo*
mmii	*miembros inferiores*
grado 2, *2do grado*, *2 grado*, *gr II*, *G2*, *GII*	*segundo grado*
hta	*hipertensión arterial*
ira	*insuficiencia renal aguda*
oma	*otitis media aguda*
AF, *antec fliar*, *atc familiar*	*antecedente familiar*
fratura	*fractura*
trauatismo	*traumatismo*
dematitis	*dermatitis*
litisis	*litiasis*
Reynaud/Raynoud	*Raynaud*
Hodkin, *Hodking*	*Hodgkin*
dolores de cabeza	*dolor de cabeza*
fallecimiento , *muerte*	*deceso*
de hígado	*hepático*
biológico	*natural*
cáncer	*neoplasia maligna*

Table 2: Examples of equivalent strings.

We use a machine learning approach to learn whether a pair of descriptions is a match or not. We create a family of features to train classification algorithms. Hyper-parameters were adjusted using 5-fold cross-validation. Models are based on different combinations of feature sets explained in next subsection.

3.4 Features

Features are organized in sets $S1, ..., S10$ and then different set combinations are used to generate the corresponding models. In Table 3 d_1 and d_2 are the description strings that are compared, where d_1 vectors represent queries and d_2 a retrieved string. The training corpus was used to adjust the corresponding d_2 vectors, using both word unigrams and character bi-grams.[4]

Feature set $S1$ represents string metrics to obtain differences in string characteristics between a pair of description terms. Features in $S2, S3, S4, S5$ are traditional representations in vector space model of d_1 and d_2 based on unigram word and bi-gram character representation with TF-IDF, binary occurrence and term frequency weights. In S_6 and S_7 we consider differences in descriptions (d_{12} and d_{21}), and S_9, S_{10} considers context (c) also.

We define $w(d)$ as the set of words in d, and d_{12}, d_{21} and c as follows

$$d_{12} = w(d_1) \setminus w(d_2)$$

$$d_{21} = w(d_2) \setminus w(d_1)$$

Set	**Feature**
S1	$L_1 = length(d_1)$ $L_2 = length(d_2)$ $m = min(L_1, L_2)$ $M = max(L_1, L_2)$ $ratio_{length} = \frac{m}{M}$ $difference_{length} = \|M - m\|$ $Levenshtein_{ratio}(L_1, L_2)$ $Jaccard(L_1, L_2)$
S2	Vector of unigram word occurrence in d_1 Vector of unigram word occurrence in d_2
S3	Vector of unigram word TF-IDF in d_1 Vector of unigram word TF-IDF in d_2
S4	Vector of bigram character frequency in d_1 Vector of bigram character frequency in d_2
S5	Vector of bigram character TF-IDF in d_1 Vector of bigram character TF-IDF in d_2
S6	Vector of unigram word occurrence in d_{12} Vector of unigram word occurrence in d_{21}
S7	Vector of bigram character frequency in d_{12} Vector of bigram character frequency in d_{21}
S8	Vector of unigram word occurrence in d_{12} Vector of unigram word occurrence in d_{21} Vector of unigram word occurrence in c
S9	Vector of bigram character frequency in d_{12} Vector of bigram character frequency in d_{21} Vector of bigram character frequency in c
S10	Vector of group of words in d_{12} Vector of group of words in d_{21} Vector of group of words in c

Table 3: Feature-sets.

$$c = w(d_1) \cap w(d_2)$$

For example:

d_1 = fractura de rodilla izquierda,
d_2 = fractura de rodilla izq then
$w(d_1)$ = $\{$*fractura*, *de*, *rodilla*, *izquierda*$\}$,
$w(d_2)$ = $\{$*fractura*, *de*, *rodilla*, *izq*$\}$,
d_{12} = $\{$*izquierda*$\}$
d_{21} = $\{$*izq*$\}$ and c = $\{$*fractura, de, rodilla*$\}$

3.5 Unsupervised Learning of Synonyms, Abbreviations and Typos

In this section we present an approach to detect word synonyms, abbreviations, acronyms and frequent typographical errors. We explain how the set of features $S10$ was generated.

Unsupervised algorithms were studied widely in the literature to detect relationships between words in order to improve results of NLP tasks such us chunking or named entity recognition. Clustering to detect word equivalence classes from unlabeled corpus were studied in (Kneser and Ney, 1993) and (Turian et al., 2010).

We introduce a procedure to generate sets of semantically equivalent strings from term

[4] Features d_{12} and d_{21} are explained below.

descriptions using a graph algorithm.

Given a set of positive description pair matchings such as

(15) d_1: sospecha de infección urinaria
d_2: probable infección urinaria

(16) d_1: urticaria en cara
d_2: urticaria en rostro

(17) d_1: duelo por fallecimiento de padre
d_2: duelo por muerte de padre

(18) d_1: duelo por deceso de padre
d_2: duelo por muerte de padre

The following semantically equivalent pairs can be inferred using word differences between pairs of descriptions:

$\{$*sospecha*, *probable*$\}$, $\{$*cara*, *rostro*$\}$ $\{$*fallecimiento*, *muerte*$\}$ and $\{$*deceso*, *muerte*$\}$

Therefore it is possible to replace, for example, the terms *sospecha* and *probable* by a concept representing this class with some label. Using the concept class label instead of a term as a feature in a vector space model we can deal with synonymy problems.

Since this approach only infers direct associations, we cannot detect the pair $\{$*deceso*, *muerte*$\}$ using this approach.

Semantically equivalent pairs can be extended to larger sets (semantically equivalence classes), building an undirected weighted graph, considering terms as vertices and equivalent pairs as edges. Connected components in the graph can be detected and terms can be clustered in some cases.

An undirected weighted graph $G = (V, E, W)$ is generated creating an edge $(d_{12}, d_{21}, w) \in E$ for each pair of descriptions d_1, d_2 such that $\mid d_{12} \mid=\mid d_{21} \mid= 1$. For example, the pair of descriptions *duelo por fallecimiento de padre* and *duelo por muerte de padre* generates the *fallecimiento* and *muerte* connection. In the same way, *deceso* and *muerte* are connected. The weight associated with each edge is the frequency of the corresponding pair in T_1.

The graph constructed under this approach is composed of different connected components. Figure 3 shows some connected components in the final graph once all edges are generated using T_1 and considering only edges with minimum frequency of 20 (lower frequency thresholds are very sensible to noisy data while higher values results in loss of information). Vertices in the same connected component are potentially equivalent. The connected components of G can be computed in linear time using either depth-first search or breadth-first search approach.

Since some terms can be ambiguous, they can be connected to some non-equivalent terms, like *od* which can be connected to *ojo derecho* (*right eye*) and *oido derecho* (*right ear*). In those cases, the connected component containing an ambiguous term, includes more than one concept. In a vector space model, in some cases disambiguation can be obtained from the context. For example in *otitis od* the *od* term refers to *oido*, while in *conjuntivitis od* refers to *ojo*. It would still be desirable to partition the connected component breaking edges like *ojo derecho* and *oido derecho*.

We used the label propagation algorithm described in (Raghavan et al., 2007). It is a clustering algorithm intended to be applied in social communities detection in large-scale networks and biochemical networks among other domains. This

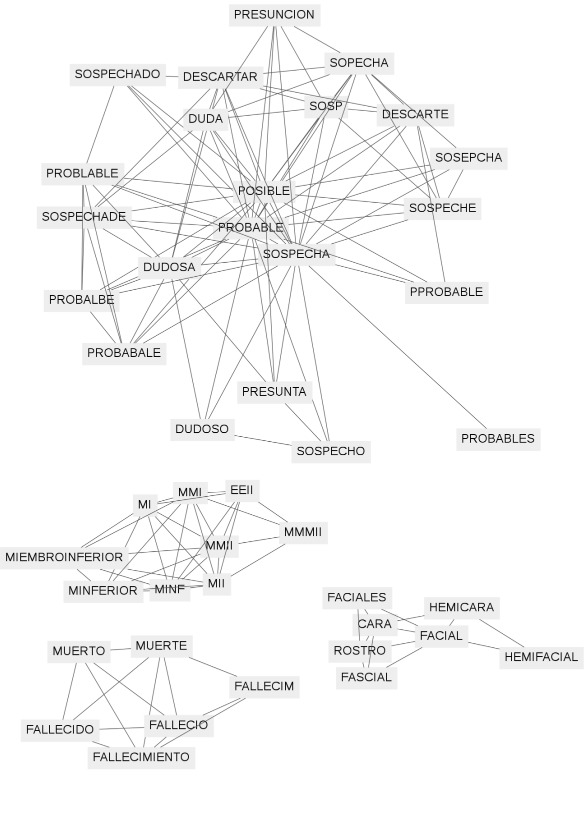


Figure 3: Word graph connected components example

clustering algorithm computes clusters based on the network structure and -unlike other approaches like k-means or DBSCAN- there is no requirement to specify the number of clusters or the neighbourhood size as parameters. The algorithm initializes each node with a unique identifier, and iteratively assigns to each node the label that most of its neighboors currently have. We run this label propagation algorithm to obtain a clustering analysis on large connected components that contained different word meanings.

As a final example, combining different terms from the final set of equivalence classes, we obtain a unique representation of the following 72 possible ways to express *duelo por fallecimiento de padre biológico debido a cáncer renal*:

$$\text{Duelo por} \left\{ \begin{array}{c} \text{fallecimiento} \\ \text{muerte} \\ \text{deceso} \end{array} \right\} \text{de padre} \left\{ \begin{array}{c} \text{biológico} \\ \text{natural} \end{array} \right\}$$

$$\left\{ \begin{array}{c} \text{debido a} \\ \text{a causa de} \end{array} \right\} \left\{ \begin{array}{c} \text{cáncer} \\ \text{ca} \\ \text{neoplasia maligna} \end{array} \right\} \left\{ \begin{array}{c} \text{de riñón} \\ \text{renal} \end{array} \right\}$$

4 Experiments and Results

Our experiments were conducted by using *scikit-learn* machine learning library (Pedregosa et al., 2011) with *liblinear* (Fan et al., 2008) solver for MaxEnt, considering L2 regularization. Hyperparameter C was determined by 5-fold cross-validation considering F1 measure. We trained XGBoost model, with binary logistic objective and F1 score as evaluation metric, by using XGBoost library described in (Chen and He, 2015). Connected components in graph and label propagation algorithm for graph clustering were conducted by using *igraph* library.

In order to generate word equivalence classes for $S10$ we found 278,555 concepts in the thesaurus with at least two associated descriptions which generate 5,956,368 potential pairs of descriptions connected to the same concept. Filtering pairs of descriptions such that both shares the same words except one, we obtain 505,447 word associations. By taking the connected components of G we get $505,447$ edges and 805 groups. Finally, clustering connected components for which more than one meaning are represented, we obtain 4,711 words in 957 group of words.

We compare the predictive power to classify a pair of descriptions as a positive match by calculating the *F1* measure on different models. Also, we compare the ability to rank the retrieved results using the classification model probability as scoring by calculating P@1, R@1 and the mean reciprocal rank (*MRR*).

By using IR score with some fixed threshold we define a classifier algorithm with its respective precision and recall (as threshold increases, recall decreases and precision increase). Figure 4 shows IR score precision-recall curve against string metrics features based fitted models. Table 4 shows MaxEnt and XGBoost F1 score for string features based models.

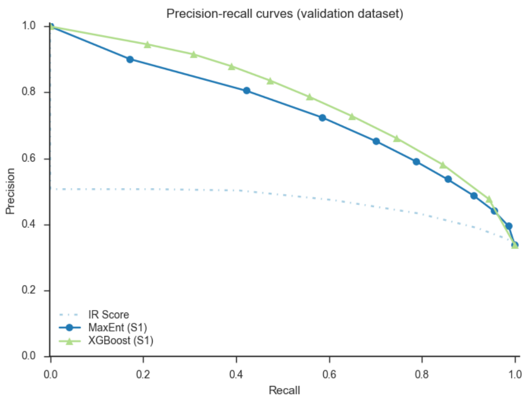


Figure 4: Precision-recall curves (String metrics features)

IR comparison vs MaxEnt and XGBoost models based on string metrics features.

Featureset	Source	MaxEnt	XGBoost
String metrics (S1)	d_1, d_2	0.67	0.70

Table 4: MaxEnt and XGBoost F1-score over string metrics

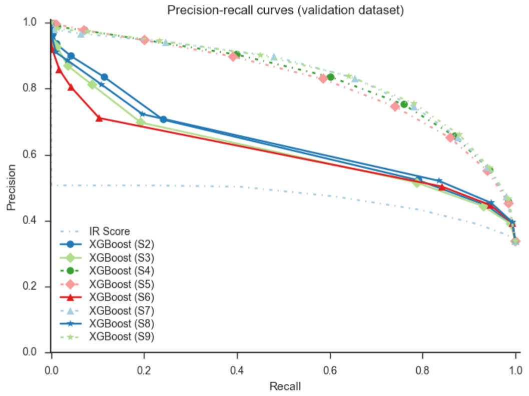
By considering F1 measure on string metrics ($S1$) and vector space model representation of descriptions ($S2, S3, S4, S5$), XGBoost showed a considerable improvement on bi-gram character features based (see Table 5) either on frequency ($S4$) or TF-IDF ($S5$) weight schemas, outperforming MaxEnt.

	Source	Weight	MaxEnt	XGBoost
(S4)	d_1, d_2	freq.	0.57	0.76
(S5)	d_1, d_2	tf-idf	0.56	0.74
(S7)	d_{12}, d_{21}	freq.	0.58	0.76
(S9)	d_{12}, d_{21}, c	freq.	0.72	0.77

Table 5: MaxEnt and XGBoost F1-score over bigram character features $S4, S5, S7, S9$

Each XGBoost bi-gram character features based model (dashed lines with markers) outperforms the word features based models (solid lines). Precision values are given across all recall levels (Figure 5).

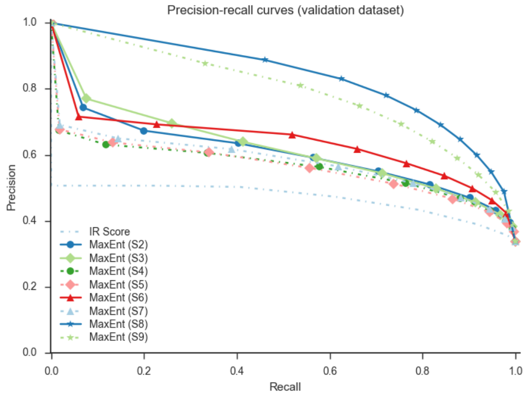
Figure 5: Precision-recall curves (XGBoost)



Markers are present on word and bi-gram features curves. IR curve has no markers. Word features are represented by solid lines while bi-gram character features are represented in dashed lines. Marker type indicates a specific source, for example $S6$ and $S7$ triangle correspond to d_{12}, d_{21} source.

s

Figure 6: Precision-recall curves (MaxEnt)



Markers are present on word and bi-gram features curves. IR curve has no markers. Word features are represented by solid lines while bi-gram character features are represented in dashed lines. Marker type indicates a specific source, for example $S6$ and $S7$ triangle correspond to d_{12}, d_{21} source.

We can see in Figure 6 that word features based models improves performance over bi-gram character feature based models using MaxEnt (each source is represented by a marker type, e.g. a triangle for the source d_{12}, d_{21}). $S8$ and $S9$ features outperform the others features on MaxEnt. Results on word features are detailed in Table 6.

	Source	Weight	MaxEnt	XGBoost
(S2)	d_1, d_2	binary	0.59	0.59
(S3)	d_1, d_2	tf-idf	0.59	0.58
(S6)	d_{12}, d_{21}	binary	0.63	0.62
(S8)	d_{12}, d_{21}, c	binary	0.76	0.62

Table 6: MaxEnt and XGBoost F1-score over unigram word features $S2, S3, S4, S6, S8$

With respect to the set of features considering word difference between pairs of descriptions ($S6, S7$), XGBoost also performs the task better when consider bi-gram character features ($S7$) as shown in Table 5, while MaxEnt works better on word features ($S6$) as shown in Table 6. When context vector is present along with word difference representation ($S6$ vs $S8$ and $S7$ vs $S9$), MaxEnt showed a considerable improvement in $S8$ respect to $S6$ (see Table 6) but XGBoost achieves a slightly improvement in $S9$ compared to $S7$ (see Table 5, $S6$ vs $S8$ and $S7$ vs $S9$). Word difference vector representation worked better in MaxEnt, than combining string metric and traditional word or n-gram based representation of descriptions, while XGBoost achieves similar performance when consider that combination.

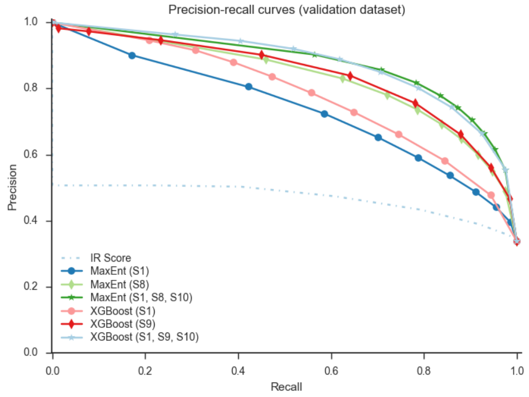
When word equivalence classes features based models are considered ($S10$), MaxEnt and XGBoost achieves similar performance (see Table 7).

Featureset	Source	MaxEnt	XGBoost
(S10)	d_1, d_2	0.69	0.69

Table 7: MaxEnt and XGBoost F1-score over word equivalence class features

By combining ($S1$, $S8$, $S10$) features MaxEnt achieves an F1 score of 0.87, while XGBoost achieves an F1 score of 0.86 by combining ($S1$, $S9$, $S10$) as showed in Table 8 and Figure 7 improving the previous models.

Figure 7: Precision-recall curves



MaxEnt vs XGBoost comparison. Circle markers represent string based features ($S1$) models, diamond d_{12}, d_{21}, c based models. Models combining string, d_{12}, d_{21}, c and $S10$ features are represented by curves with star markers.

Model	Prec	Rec	F1
IR	0.73	0.76	0.74
MaxEnt (S1, S8, S10)	0.80	0.95	0.87
XGBoost (S1, S9, S10)	0.77	0.96	0.86

Table 8: MaxEnt and XGBoost F1-score over feature sets combination

To evaluate these models performance on ranked results, we compute the P@1, R@1 and mean reciprocal rank (MRR) metrics showed in Table 9.

Model	P@1	R@1	F1	MRR
IR	0.73	0.76	0.74	0.84
MaxEnt (S1, S8, S10)	0.87	0.91	0.89	0.94
XGBoost (S1, S9, S10)	0.87	0.91	0.89	0.94

Table 9: MaxEnt and XGBoost F1-score over feature sets combination

5 Conclusions and future work

We presented a hybrid Soft-TFIDF and machine learning approach to bio-medical terms normalization. This technique can be used in different problems such as automatic coding of descriptions and reference resolution in general. Our approach neither requires any additional resource like acronyms/abbreviations, alias and synonyms lists nor a spell checker because that ability is acquired from examples by defining a scoring function learned from data. As a result, our approach shows very good F1 score and mean reciprocal rank results. Even though the data set was in Spanish, we did not use any specific resource for that language, therefore our approach can be replicated in any language.

Creation of new features based on differences between descriptions and its context, in addition to the more traditional features, allow machine learning models to improve detection of pairs of semantically equivalent descriptions with low syntactic similarity and discard non semantically equivalent ones with high syntactic similarity by learning semantic equivalence from pairs of descriptions examples. As result, the false negative and false positive rates were reduced.

By generating a clustering of words to find synonyms, specially from indirect associations between words from descriptions across different concepts from direct associations, the semantic feature space generated improved the performance of machine learning models increasing F1 measure.

Finally, MaxEnt and XGBoost models showed to be effective for the task with some minor differences in the set of features returning best results.

Our work was based on the performance of the text search engine results. Then, this approach can not consider results that were not retrieved by the search engine. To overcome this limitation it is possible to use a query expansion approach. Alternatively, words in the terms can be transformed to a canonical form, both at index and query time. We also plan to expand this work to other bio-medical domains such as procedures or drugs.

References

Alan R Aronson and François-Michel Lang. 2010. An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236.

Mikhail Belenko and Raymod J. Mooney. 2003. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Datamining*, pages 39–48, Washington D.C.

Blaschke C., Hirschman L., Yeh A., and Valencia A. 2003. Critical assessment of information extraction systems in biology. *Comparative and Functional Genomics*, pages 674–677.

J. Castaño, J. Zhang, and J. Pustejovsky. 2002. Anaphora resolution in biomedical literature. In *International Symposium on Reference Resolution*, Alicante, Spain.

Tianqi Chen and Tong He. 2015. Higgs boson discovery with boosted trees. In *Cowan et al., editor, JMLR: Workshop and Conference Proceedings*, number 42.

William Cohen and Jacob Richman. 2001. Learning to match and cluster entity names. In *ACM SIGIR-2001 Workshop on Mathematical/Formal Methods in Information Retrieval*, New Orleans, LA, September.

William W. Cohen, Pradeep Ravikumar, and Stephen Fienburg. 2003. A comparison of string metrics for matching names and records. In *KDD Workshop on Data Cleaning and Object Consolidation*.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Haw-ren Fang, Kevin Murphy, Yang Jin, Jessica S. Kim, and Peter S. White. 2006. Human gene name normalization using text matching with automatically extracted synonym dictionaries. In *Proceedings of the Workshop on Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis*, BioNLP '06, pages 41–48, Stroudsburg, PA, USA. Association for Computational Linguistics.

Carol Friedman, Lyudmila Shagina, Yves Lussier, and George Hripcsak. 2004. Automated encoding of clinical documents based on natural language processing. *Journal of the American Medical Informatics Association*, 11(5):392–402.

Maria Laura Gambarte, Alejandro Lopez Osornio, Marcela Martinez, Guillermo Reynoso, Daniel Luna, and Fernan Gonzalez Bernaldo de Quiros. 2007. A practical approach to advanced terminology services in health information systems. *Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics; Building Sustainable Health Systems*, page 621.

Ning Kang, Bharat Singh, Zubair Afzal, Erik M van Mulligen, and Jan A Kors. 2013. Using rule-based natural language processing to improve disease normalization in biomedical text. *Journal of the American Medical Informatics Association*, 20(5):876–881.

Rohit J Kate. 2016. Normalizing clinical terms using learned edit distance patterns. *Journal of the American Medical Informatics Association*, 23(2):380–386.

Jung-Jae Kim and Jong C. Park. 2004. Bioar: Anaphora resolution for relating protein names to proteome database entries. In Sanda Harabagiu and David Farwell, editors, *ACL 2004: Workshop on Reference Resolution and its Applications*, pages 79–86, Barcelona, Spain, July. Association for Computational Linguistics.

Reinhard Kneser and Hermann Ney. 1993. Forming word classes by statistical clustering for statistical language modelling. In *Contributions to Quantitative Linguistics*, pages 221–226. Springer.

Hanna Köpcke and Erhard Rahm. 2010. Frameworks for entity matching: A comparison. *Data & Knowledge Engineering*, 69(2):197 – 210.

Robert Leaman, Rezarta Islamaj Doan, and Zhiyong Lu. 2013. Dnorm: disease name normalization with pairwise learning to rank. *Bioinformatics*, 29(22):2909–2917.

Y. Lin and T. Liang. 2004. Pronominal and sortal anaphora resolution for biomedical literature. In *Proceedings of ROCLING XVI: Conference on Computational Linguistics and Speech Processing*, Taipei, Taiwan.

Alejandro Lopez Osornio, Daniel Luna, Maria Laura Gambarte, Adrian Gomez, Guillermo Reynoso, and Fernan Gonzalez Bernaldo de Quiros. 2007. Creation of a local interface terminology to snomed ct. *Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics; Building Sustainable Health Systems*, page 765.

Erwan Moreau, François Yvon, and Olivier Cappé. 2008. Robust similarity measures for named entities matching. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 593–600, Stroudsburg, PA, USA. Association for Computational Linguistics.

Serguei VS Pakhomov, James D Buntrock, and Christopher G Chute. 2006. Automating the assignment of diagnosis codes to patient encounters using example-based and machine learning techniques. *Journal of the American Medical Informatics Association*, 13(5):516–525.

Jon Patrick, Yefeng Wang, and Peter Budd. 2006. Automatic mapping clinical notes to medical terminologies. In *Proceedings of the 2006 Australasian Language Technology Workshop (ALTW 2006)*, pages 75–82.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noémie Elhadad. 2014. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association*, 21(2):231–237.

J. Pustejovsky, J. Castaño, B. Cochran, M. Kotecki, and M. Morrell. 2001. Automatic extraction of acronym-meaning pairs from medline databases. In *Proceedings of Medinfo, London*.

Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106.

S Trent Rosenbloom, Randolph A Miller, Kevin B Johnson, Peter L Elkin, and Steven H Brown. 2006. Interface terminologies: facilitating direct entry of clinical data into electronic health record systems. *Journal of the American Medical Informatics Association*, 13(3):277–288.

Ale Rudniy, Min Song, and James Geller. 2014. Mapping biological entities using the longest approximately common prefix method. *Bioinformatics*, 15.

Mary H Stanfill, Margaret Williams, Susan H Fenton, Robert A Jenders, and William R Hersh. 2010. A systematic literature review of automated clinical coding and classification systems. *Journal of the American Medical Informatics Association*, 17(6):646–651.

Yao Sun. 2004. Methods for automated concept mapping between medical databases. *Journal of Biomedical Informatics*, 37(3):162 – 178.

Hanna Suominen, Filip Ginter, Sampo Pyysalo, Antti Airola, Tapio Pahikkala, S Salanter, and Tapio Salakoski. 2008. Machine learning to automate the assignment of diagnosis codes to free-text radiology reports: a method description. In *Proceedings of the ICML/UAI/COLT Workshop on Machine Learning for Health-Care Applications*.

Yoshimasa Tsuruoka, John McNaught, Jun'i;chi Tsujii, and Sophia Ananiadou. 2007. Learning string similarity measures for gene/protein name dictionary look-up using logistic regression. *Bioinformatics*, 23(20):2768–2774.

Yoshimasa Tsuruoka, John McNaught, and Sophia Ananiadou. 2008. Normalizing biomedical terms by minimizing ambiguity and variability. *BMC Bioinformatics*, 9(3):1–10.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Tielman T Van Vleck, Adam Wilcox, Peter D Stetson, Stephen B Johnson, and Noémie Elhadad. 2008. Content and structure of clinical problem lists: A corpus analysis.

Ben Wellner, José Castaño, and James Pustejovsky. 2005. Adaptive string similarity metrics for biomedical reference resolution. In *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*, ISMB '05, pages 9–16, Stroudsburg, PA, USA. Association for Computational Linguistics.

Joachim Wermter, Katrin Tomanek, and Udo Hahn. 2009. High-performance gene name normalization with geno. *Bioinformatics*, 25(6):815–821.

H. Yu and E. Agichtein. 2003. Extracting synonymous gene and protein terms from biological literatur e. *Bioinformatics*, pages 340–349.