DNER Clinical (named entity recognition) from free clinical text to Snomed-CT concept

IGNACIO MARTINEZ SORIANO, JUAN LUIS CASTRO PEÑA
Computation Science and Artificial Intelligence
University of Granada
C/Daniel Saucedo Aranda s/n, 18071 Granada
SPAIN
Ignacio.martinez@carm.es, castro@decsai.ugr.es http://decsai.ugr.es/~castro

Abstract: We have developed a new approach for the (NER) named entity recognition problem, in specific domains like the medical environment. The main idea is recognize clinical concepts in free text clinical reports. Actually most of the information contained in clinical reports from the Electronic Health System (EHR) of a hospital, is written in natural language free text, so we are researching the problem of automatic clinical named entities recognition from free text clinical reports, in this kind of texts we design a new NER approach, like a hybrid of theses approach, dictionary-based, machine learning, and a fuzzy function. To develop this, from clinical reports free text, we apply an unsupervised, shallow learning neural network, word2vec to represent words of the text as "words vectors". Second, we use a specific domain dictionary-based gazetteer (using the ontology Snomed-CT to get the standard clinical code for the clinical concept), for match the correct concept, and recognize the named entity like a clinical concept, we use the distance and similarity between of the "words vector" of the terms from the document and the distance of the "word vector" with the Snomed-CT description term, applying a fuzzy function "DNER", to get the best degree of identification for the named entity recognized. We have applied our approach on a Dataset with 318.585 clinical reports in Spanish from the emergency service of the Hospital "Rafael Méndez" from Lorca (Murcia) Spain, and preliminary results are encouraging.

Key-Words: Snomed-CT, word2vec, doc2vec, clinical information extraction, skipgram, medical terminologies, search semantic, named entity recognition, ner, medical entity recognition

1 Introduction

Electronic Health Records, include a great variety of information from different sources, depend of the structure of the EHR , this information is stored, in a structured text using a run vocabulary with a normalized clinical terminology , but the majority of the (EHR) has the clinical information unstructured as free text, and access to the knowledge inside is very hard, to the management care of patient, medical research or decision support systems.

All the Spanish hospitals need a human codification team, to assign standards codes, from a clinical terminology like (ICD-10MC), to normalize the diagnostics, and the procedures of the medical reports [21].

To do this process, is necessary discover the relevant clinical entities, from the information locked up in the free text of medical records. This job is hard and slow, depending of the experience of the human encoder. So the automatic identification of these unstructured information entities, is an important task and one of the main challenger for analysis of free-text electronic health records. Natural Language Processing (NLP) techniques provide a solution to process the identification of clinical entities with a "named entity recognition" algorithm (NER).

We propose a new NER method to identify the clinical entities in the free text from the sections of the Emergency EHR, and assign a Snomed-CT ID concept, to transform the unstructured free text of the clinical reports, to a structured set of Snomed-CT concept.

The rest of this article is organized as follows. In the next section, we identify the NER process we want to improve, in section 2.1 we describe how NER work, what is medical NER, and state of art of this. In section 2.2 we define the structure, design and implementation of the clinical terminology Snomed-CT. In Section 2.3 we explain the representation of vectors word by word2vec and section 2.4 we define doc2vec. In Section 2.5 we explain what python library we use to implement our solution. In Section

2.6 we show how visualize "word vectors". Section 3 we define the problem solution and planning, and 3.1 how we use with the clinical reports. In Section 3.2 we define the new approach DNER, using Word2Vec. And Section 4 the conclusions.

2 Problem Formulation

The use of NER-Tools, apply to a general purpose vocabulary has showed a proper functioning, Gangemi[1] compared the NER-tools using common texts and general purpose vocabulary, Hooland[2] has compared Ner-Tools using domain-specific texts and general purpose vocabulary, our work is how we can apply our new approach of NER-Tool, in a domain-specific texts and domain specific vocabulary.

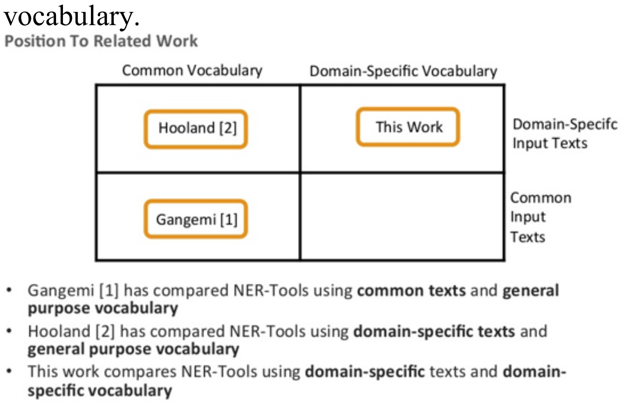

Fig. 10

Heuss[3], NER tools perform signicatively worse in domain specific scenarios. To identify the clinical entities we have seen that using the classical approach (NER) Machine Learning, did not get good results. So we want to change the approach using a Neural Network Word2Vec algorithm.

We have a huge dataset from clinical reports, and then it is a good field to use Word2Vec for the training of our Neural Network and create the correct model of vectors words to use in our NER-tool.

2.1 Named Entity Recognition

The NER process is a main task from the information extraction systems, the main target is identify all the named entities from the free text. A "named entity" is a nominal sentence or term, that identify an item from a set, with others items with similar attributes [6].

This process has two task: Entity identification, and classification of categories sets. These categories can adjust to a particular domain like the clinical (example, diseases) [10].

There are three approach to implement a (NER) Algorithm: ruler-based system, dictionary-based systems (gazetteers) and Machine Learning System. *Rule-based systems* use pattern identification techniques in the Text as heuristics derived from both, the morphology and the semantics of input phrases. Usually works like classificatory with Machine Learning approach. Systems based only in rules are likely to skip "named entities" and sometimes, exceeded in the recognition of entities. *Gazetteer-based* approaches use external knowledge resources to identify pieces of text using a dictionary or lexicon built with names of entities. This approach requires a manual work, with an expert, for the creation of the lexicon or an automatic system to build from an external resource. This kind of approach is difficult to develop, but the results are better for specific domains, like "medical domain".

The Machine Learning approach are the best solutions working with different domains, provide predictive analysis in entities that are unknown for "gazetteers".

The most used resources in this field, are the *conditional random fields* (CRF) and the *Markov hidden models*.

The CRF is a kind of statistic model, use to discover pattern, given the context of a neighborhood. Hidden Markov Model tagging generates entity tags named on the original text by calculating the probability that a word is a named entity using n-gram frequencies of a training set.

2.1.1 Medical Named Entity Recognition

In the medical domain, NER systems [11] are called Medical Entity Recognition (MER). These systems try to detect and delimit Medical entities in texts and classify it into a given category [12]. More of the MER systems use machine learning techniques (CRF) using input an annotated data set. In medical field is very usual, that appear new names of concepts and abbreviations, so the gazetteer approach is insufficient in clinical practice.

There are many models of clinical NER, one of them are:

-IxaMed: Applying Freeling and a Perceptron Sequential Tagger at the Shared Task on Analyzing Clinical Texts.[13]

-TMT: A tool to guide users in finding information on clinical texts [14].

-SZTE-NLP: Clinical Text Analysis with Named Entity Recognition [15].

-(Noble Coder)Named Entity Recognition (NER) engine for biomedical text.[16]

2.2 Snomed-CT

The main clinical terminology that we used with our NER is Snomed-CT (Systematized Nomenclature of Medicine Clinical Terms).

It is widely recognized as the leading global clinical terminology for use in electronic health records. It is maintained and developed by an International body (the IHTSDO) In Mar 2 2017 IHTSDO adopted the trading name of SNOMED International [17] to reflect our focus on the SNOMED CT product. Is the most comprehensive, multilingual clinical healthcare terminology in the world. Enables consistent, processable representation of clinical content in electronic health records.SNOMED CT based clinical information benefits individual patients and clinicians as well as populations and it supports evidence based care.

Features of Snomed-CT include, A broad scope that covers most of the clinical concepts used in patient centered clinical records;Ability to express different levels of clinical detail in patient record entries by using expressions containing one or more concept identifiers; Relationships between concepts that enable consistent retrieval of a common form of clinical information for many different purposes;A reference set mechanism to support representation of language / dialect variants, value sets, alternative hierarchies and mapping to classifications.

Components of Snomed-CT: it is define like **concepts with uniques meanings** and definitions from a formal logic by hierarchies.

- *Concepts*, represent clinical meaning organized by hierarchies.
- *Descriptions*, which link appropriate human readable terms to concepts.
- *Relationships*, link each concept each other.

Concepts and Hierarchies: Represent clinical meaning, every concept is a clinical idea associated with a unique identifier. The hierarchy is organized from general to more detailed.

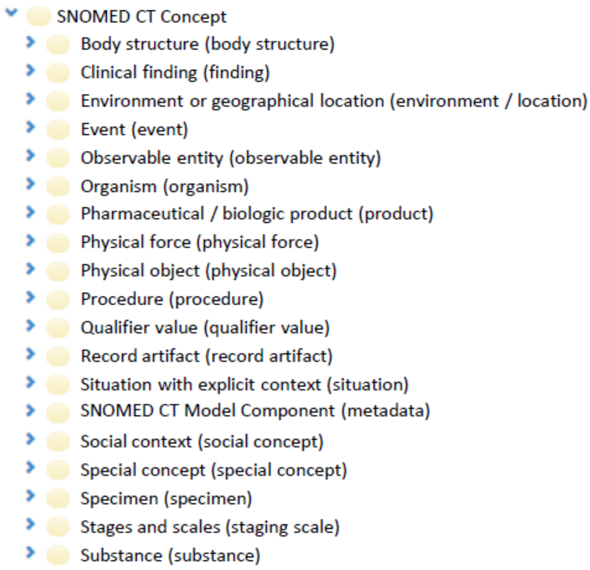

Fig.1 [18]

*Descriptions:*A description links a human-readable term to a concept. Each concept is associated with several descriptions. A concept can be different descriptions and each represent a synonym from the same clinical concept.Each description has a number identificator unique.

Terms are character strings that consist of words, phrases and other human-readable representations.Each descriptions has a *description type* and may be marked as **preferred** in particular languages.

There are two commonly used description types:**Fully Specified Name** (FSN), to uniquely describe a concept and clarify its meaning. A concept may have more than one FSN, but only one of these may be marked as **preferred in a given language**.

Synonym, represents a **term**, other than FSN, to represent a concept in a particular language or dialect.

Relations: Each concept is associated with other concepts by a set of relationships. The relationships express defining characteristics of a concept. Link concepts with other concepts, with a meaning. The structure of the relationship is like triplet (Object, Attribute, Valor). Include the *source concept*, the *identifier of the relationship* type concept **"is a**", and the identifier of the *destination* concept.

Attribute Relationships: An attribute relationships is an association between two concepts that specifies a defining characteristic of one of the concepts (the source of the relationship).

2.2.1 Logical Model.

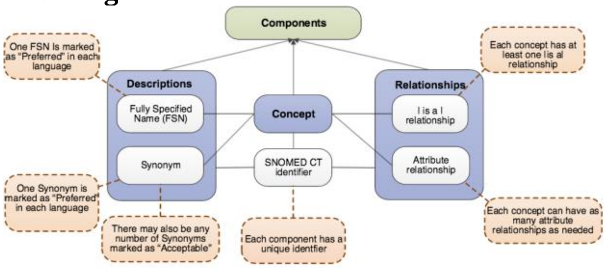

Fig.2[18]

The structure of the core logical model is:

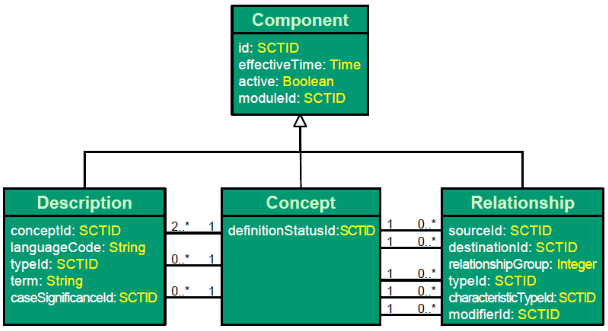

Fig.3

2.3 Representation of Word2vec Model.

Word2vec is a semantic learning framework that uses a shallow neural network to learn the representations of words in a text.The idea is would be able to learn the context of a word in a text.

This model was developed by Mikolov et al [19], produce a distributed representations of words in a vector space, grouping similar words. *"The main goal of that paper was to introduce techniques that can be used for learning high-quality word vectors from huge data sets with billions of words, and with millions of words in the vocabulary"*.

The most common techniques for representation of text was:

-Local representations (N-grams, Bag-of-words, 1-of-N coding)

- Continuous representations (Latent Semantic Analysis, Latent Dirichlet Allocation, Distributed Representations). Using different neural network based language models, define distributed representations of words, like Feed forward neural net language model or recurrent neural net language model. A neural network language model (NNLM) arquitecture was suggest in [20], where a feed forward neural network with a linear projection layer and a non-linear hidden layer was used to learn the word vector representation and a statistical model.

word2vec is not a single algorithm, it is a software package for representing words as vectors, this module containing:

- Two distinct models
    - CBoW(continuous bag-of-words)
    - Skip-Gram
- Various training methods
    - Negative Sampling
    - Hierarchical Softmax
- A rich preprocessing pipeline
    - Dynamic Context Windows
    - Subsampling
    - Deleting Rare Words

2.3.1 CBoW (Continuous bag-of-words)

This model assume that there is only one word considered per context, it will predict one target word given one context word.

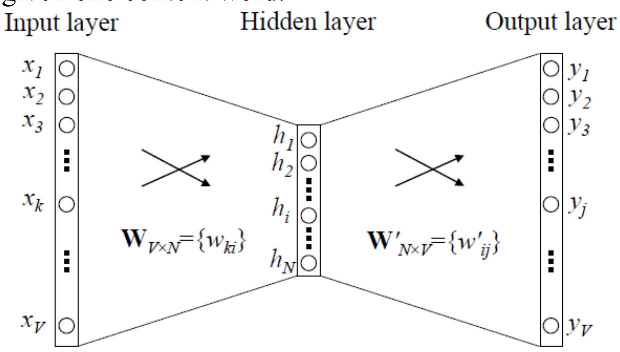

Fig.4

Our setting, the vocabulary size is V, and the hidden layer is N, the nodes on adjacent layers are fully connected. The input vector is a one-hot encoded vector, that only one node of $\{x_1,\ldots,x_v\}$ will be 1, and all other nodes are 0. The weights between the input layer and the output layer is represented by a V x N matrix W. Each row of W is the N-dimension vector representation v_w.

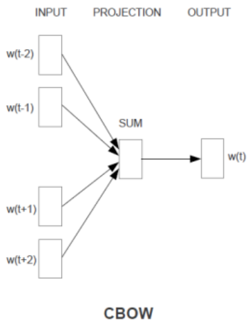

Fig.5 [21]

Predicts the current word given the context

2.3.2 Skip-Gram

The idea described in [21] this model,

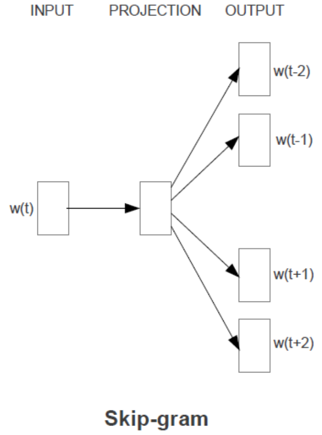

Fig.6
Predicts the surrounding words given the current word

2.4 Doc2Vec

In this model [4], propose *Paragraph Vector,*an unsupervised framework that learns continuous distributed vector representations for pieces of texts. In this model the vector representation is trained to be useful for predicting words in a paragraph.The paragraph vector is concatenate with several word vectors from a paragraph and predict the follow word in a given context. The paragraph and the word vector are trained with the stochastic gradient descent and backpropagation[5]. The paragraph vectors are unique in the midst of paragraph and the word vector are shared, when prediction time the paragraph vectors are inferred by fixingthe word vectors and training the new paragraph vectoruntil convergence.
Paragraph Vector is capable of constructing representations of input sequences of variable length.

2.4.1 Algorithm Doc2Vec

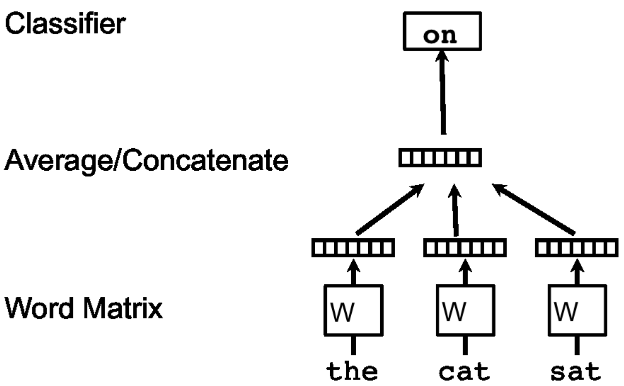

Fig.7 [4]
A framework for learning word vectors. The context of three words ("the," "cat," and "sat") is used to predict the fourth word ("on").

2.4.2 Paragraph Vector: A distributed memory model.

The idea is how the word vectors are asked to contribute to a prediction task about the next word in the sentence. So the paragraph vectors are also asked to contribute to the prediction task of the next word given many contexts sampled from paragraph. Every paragraph is mapped to a unique vector, and every word is mapped to a unique vector. Tis combination is used to predict the next word in a context.

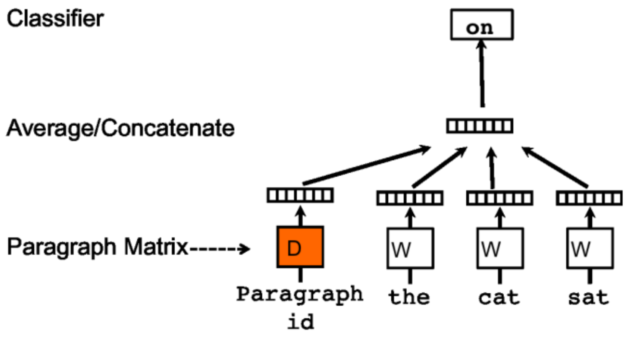

Fig.8[4]
the only change is the additional paragraph token that is mapped to a vector via a matrix D. In this model, the concatenation or average ofthis vector with a context of three words is used to predict the
fourth word. After training our paragraph vector, we can feed theses features to machine learning techniques, like logistic regression, support vector machines or K-means. The name of this approach is *Distributed Memory version of paragraph vector* (PV-DM).

2.4.3 Paragraph Vector without word ordering: Distributed bag of words.

Other way is to ignore the context words in the input, forcing the model to predict words randomly sampled from the paragraph in the output. This means that at each iteration of stochastic gradient descent, we sample a text window and form a classification task given the paragraph vector. The name of this version is *Distributed Bag of Words version of paragraph vector* (PV-DBOW).

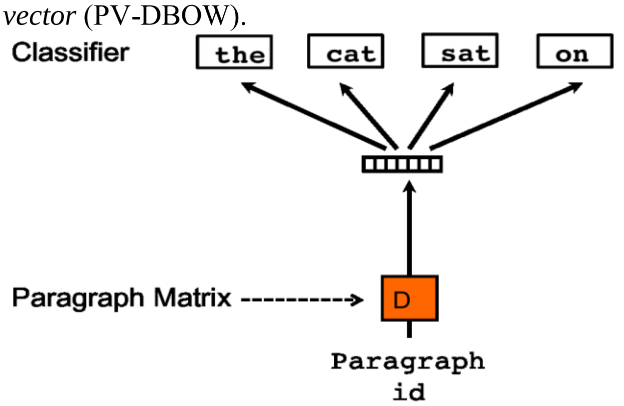

Fig.9 [4]

In this version, the paragraph vector is trained to predict the words in a small window. In this version we need store less data, only the softmax weigths. The model is similar like Skip-Gram model of word vectors [1]

2.5 Gensim a free python library

To develop our NER tool and use word2vec and doc2vec, we use this python library [7].This library Allow us, define the models Word2Vec, and Doc2vec, and depend of the parameters choose "skip-gram and CBOW models", using either hierarchical softmax or negative sampling.

2.5.1 class genism word2vec.

An example of the main parameters of class gesim training model is this[7]:
Class gensim. models. word2vec. Word2Vec (size=100, window=5, min_count=5, workers=3, sg=0, hs=0, negative=5) *Size* is the dimensionality of feature vectors. *Window* is the maximum distance between the current and predicted word within a sentence. *Min_count*, ignore all words with total frequency lower than this. *Workers* is used to many worker threads to train the model. *Sg,* define the training algorithm, by default (sh=0), CBOW is used, if sg=1, you change to skip-gram model. *hs*= if 1, hierarchical softmax will use to train the model, if (hs=0) and *negative* is non-zero, negative sampling is used. *Negative*= if is greater than 0, we use negative sampling, to specifies how many "noise words" should be drawn (usually 5-20)

2.5.2 class genism doc2vec.

The main parameter to define the class model doc2vec is[7]:
class gensim.models.doc2vec.Doc2Vec
(documents, size=100, window=8, min_count=5, workers=4, dm=1,dbow_words=0)
Documents, is the dataset, that previously we generating with the class *LabeledSentence*
from: gensim.models.doc2vec.TaggedDocument
generating a dataset, with (words, tags), Create new instance of Tagged Document(words, tags).

2.6 Visualization High Dimension Data.

To visualize a high-dimensional dataset, like word vector produce by word2vec model, we use (t-SNE) [22] t-Distributed Stochastic Neighbor Embedding , is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets by giving each data point a location in a two or three-dimensional map. This is a variation of Stochastic Neighbor Embedding [23] and produces better visualizations by reducing the tendency to crowd points together in the center of the map. For visualizing the structure of very large data sets, t-SNE can use random walks on neighborhood graphs to allow the implicit structure of all of the data to influence the way in which a subset of the data is displayed.
We use t-SNE scikit-learn[8] implementation to represent visually the insight relation between the words in the clinical reports and the concepts of Snomed-CT.
Examples of a Clinical Report representation, and a Model Simliraty of a concept.

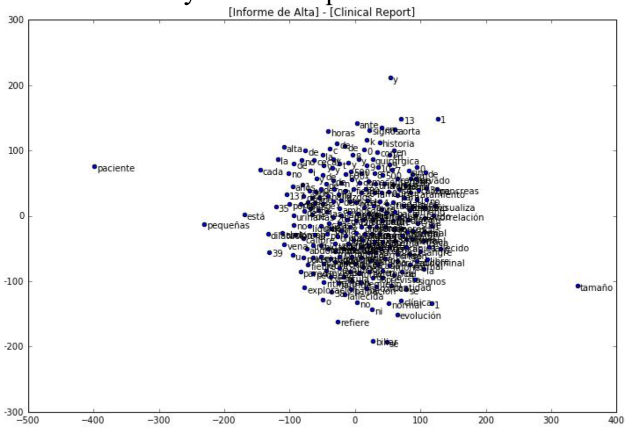

Fig. 11. Clinical Report Repesentation dots.

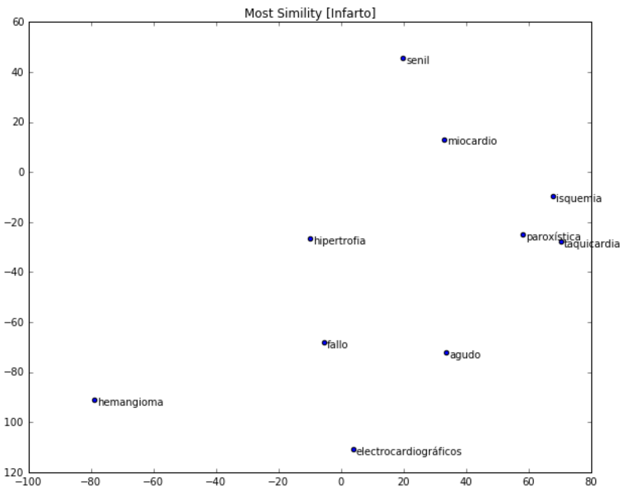

Fig.12. Similarity Clinical Concept.

3 Problem Solution

Analyzing the three kind of (NER) models, ruled-based system, gazetteer-based and Machine Learning, we see that a good approach is get a new model, with the combination of the three where we use gazetter-basel model with Snomed-CT, and machine Learning, with shallow neural networks (word2vec).

3.1 process tasks, plan

To develop our solution we need two phases:

1. NLP process, generating dataset, from the emergency clinical reports and apply (doc2vec) to generate a model of continuous vector of words, labeling our model with the different section of the emergency clinical reports. The same process we do to descriptions items from Snomed-CT.
2. We define a function "DNER", to identify the clinical named entity recognition and get the Snomed-CT IdConcept, mapping with the correct clinical concept, from our text, using the properties of the vector of words generated by our model.

3.1.1 NLP process and modelling Text.

a) We get 318.585 emergency clinical reports, from the Hospital "Rafael Mendez". These reports has a structured schema dividing in sections, with free text in every section. The Structure of our Spanish clinical report is:

- *Administrative data*, that we anonymous with a code.
- *Reason Medical Consultation:*
- *Personal Background:*
    - *known Allergies:*
    - *Medicals:*
    - *Surgeries:*
    - *Treatment Background:*
- *Actual Illness:*
- *Exploration:*
- *Complementary Evidence:*
- *Evolution:*
- *Diagnostic:*
- *Treatmenty Recommendations*:

We need to preprocess this structure with it free text, to prepare the real dataset, that we get to generate a Doc2vec Model using the genism tools. We choose this approach, because the representation of the vector to get the similarity of the words, is better if we keep in mind the sections structure of the clinical reports.

We use python nltk toolkit[9], to tokenize, stem PoS, and use Stop-words, with Spanish language.

The process task is: The first step is, create a line, with separated words, from every section of the clinical report, and assign a *Tag* with name of the section. Then we create a structured data model with *"SentenceLineLabel"* from genism, doc2vec.model, a data set with a column with ll the lines with separated words by space, and other column, with the corresponding label "tag", identifying each section of the clinical report, and then we can train our model. The main parameters that we choose to generate our model, was:

-dm=0, to use distributed bag of words (PV-DBOW), this is the same that use in Word2Vec, Skip-gram.

-windows= 10 (maximum distance between the predicted word and context words used for prediction within a document)

-min_count = 5(ignoring all words with total frequency lowe than this.

-negative=5 (if the value is >0, we use *negative sampling* for specifies how many "noise words" should be drawn).

After this we have create a vocabulary with the continuous word vector, and we can use for the posterior task.

This same process we do with the Description term, from Snomed-CT. we need to take care, that description of the concept in Snomed-CT, has a attribute that define the description class, (Fully Specified Name), (synonym), (Prefered term). But all of this identify the same Clinical Concept. In our example we use doc2vec model, and the *tag* items use the DescriptionID.

3.2 Fuzzy Named Entity Recognition. Function DNER.

In this section we describe the function to get the best candidate from description term, of Snomed-CT, and map the DescriptionID, to find the clinical concept.

Explication: We define a new function DNER to identify the "Named Entity Recognition". DNER(t,P) is a model for "degree in wich the term **t** is named in the phrase **P**".

Let **t**, a term of Snomed-CT description. With every phrase in the clinical reports sections, we identify **P** like example "Patient with infarction of H.". Choose candidates terms from descriptions:

{C1:Patient, C2:Infarction, C3:H., C4:Patient with infarction, C5:Infarction of H.}

Get the distance from **t** to **P** = "minimum distance from **t** to **C**$_i$. **DNER(t,P)** will be modeled by a Fuzzy concept "distance from **t** to **P** is small".

3.2.1. Function DNER Explication:

Given a word w, we will denote $v(w)$ the vector of the word w in the model obtained applying word2vec to all the documents.

We extend the model to any n-gram

$$if\ p = w_1 w_2 \ldots w_n\ is\ a\ ngram,$$

$$v(p) = v(w_1) + v(w_2) + \cdots + v(w_n)$$

Now we can define the distance d(g,t) between two grams g and t:

$$d(g,t) = |v(g) - v(t)|,$$

Given a paragraph $P = w_1 \ldots w_k$, we will denote$G_n(P)$ (the set of all n-grams of P with length less or equal than n:

$$G_n(P) = \{g = s_1 \ldots s_m \,/\, m \leq n\ and\ \ g\ in\ P\}$$

Now, we can define the distance of any term t (word or n-gram) to a paragraph P as the minimal distance between t and any n-gram of P:

We define

$$dist_n(t,P) = \ min\{d(t,g)/g \in G_n(P)\}$$

And finally we can define the degree in which a term t is named in a paragraph P by:

$$DNER(t,P) = \begin{cases} 0\,, & if\ dis(t,P) > M \\ \dfrac{M - dist(t,P)}{M}, & otherwise \end{cases}$$

In this way:

- DNER(t,P)=0 if all grams from P have a distance with t greather than M, so in this case we consider that t is no named in P.
- DNER(t,P)=1 if dist(t,P)=0, that is, if there exists a gram in P that is indistinguishable of t
- DNER(t,P) has a inverse linear dependence with dist(t,P), going form 0 to 1 when dist(t,P) go from M to 0.

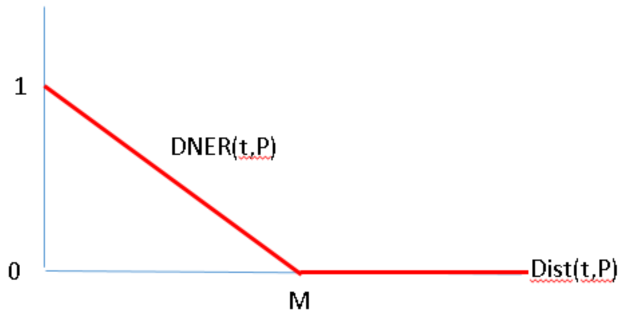

Fig.10. Representation value(M)

Finally, we will associate to any paragraph of a EHR the terms in Snomed-CT with DNER greater than a level of similarity *S*. Actually we are making experiments with different values for *n* (maximal length of n-grams) and *S* (minimal level of similarity for consider a term is named).

Summary:
We use a value for *M* to identify the threshold distance, for a named term. At begin we use an heuristic method then we are tuning by rules.
The value of *n*, is the max length of grams to choose.(ex, 3). And *S*, is the grade that we are choose to accept a named term.

To avoid a combinatorial explosion looking for all ngrams possible we are defining a set of rules that help to choose the best candidates to apply the function.

4 Conclusion

We have developed a novel approach for the named entity recognition (NER) problem in closed domains. This approach is suitable for those domains where we have a dictionary of domain concepts, and especially when we have a high amount of documents. First a word embedding model (vector representation for each word) is obtained applying wor2vec from text documents and dictionary, and the model is extended to associate a vector for any phrase, and then the distance between phrases (n-grams) of the documents and entries of the dictionary is used to recognize named entities. This approach only uses unsupervised learning in opposition with classical machine learning based NER approaches, where a great deal of manual labor is required. We have applied this approach for recognizing clinical concepts in free text clinical reports on a Dataset with 318.585 clinical reports in Spanish from the emergency service of the Hospital “Rafael Méndez” from Lorca (Murcia) Spain. The results of this preliminary experimentation are encouraging, our approach has discovered and mapped many new abbreviations and complex synonymous. Actually, we are doing a more detailed and systematic experimentation and comparison with other NER approaches.

Acknowledgements

This research has been supported by Spanish Ministry of Economy, Industry and Competitiveness, project TIN2013-48319-R “Study of Intelligent

Technologies for Monitoring Environments in Internet".

References:

[1] A. Gangemi. A Comparison of Knowledge Extraction Tools for the Semantic Web. In P. Cimiano, O. Corcho, V. Presutti, L. Hollink, and S. Rudolph, editors, The Semantic Web: Semantics and Big Data,number 7882 in Lecture Notes in Computer Science, pages 351{366. Springer Berlin Heidelberg, Jan. 2013.

[2] S. v. Hooland, M. D. Wilde, R. Verborgh, T. Steiner, and R. V. d. Walle. Exploring entity recognition and disambiguation for cultural heritage collections. Literary and Linguistic Computing, page fqt067, Nov. 2013.

[3] Timm Heuss, Bernhard Humm, Christian Henninger, and Thomas Rippl. A comparison of NER tools w.r.t. a domain-specific vocabulary. In Proceedings of the 10th International Conference on Semantic Systems (SEM '14), Harald Sack, Agata Filipowska, Jens Lehmann, and Sebastian Hellmann (Eds.). ACM, New York, NY, USA, 100-107. 2014.

[4] Quoc V Le and Tomas Mikolov, Distributed representations of sentences and document,.arXiv preprint arXiv:1405.4053., 2014.

[5] Rumelhart, David E, Hinton, Geoffrey E, and Williams, Ronald J. Learning representations by back-propagatingerrors. Nature, 323(6088):533–536, 1986.

[6] L. Ratinov and D. Roth. Design challengesand misconceptions in named entity recognition. InCoNLL, 6.2009.

[7] Radim Rehurek, Software Framework for topic Modelling with Large Corpora, Proceedings of LREC 2010 workshop on New Challenges for NLP Frameworks, 2010.

[8] Pedregosa et al. Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825-2830, 2011.

[9] Wagner, Wiebke, Steven Bird, Ewan Klein and Edward Loper. Natural Language Processing with Python, Analyzing Text with the Natural Language Toolkit - O'Reilly Media, Beijing, 2009.

[10] Jin D. Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. Introduction to thebio-entity recognition task at JNLPBA. In Proceedingsof the International Joint Workshop on NaturalLanguage Processing in Biomedicine and its Applications, JNLPBA '04, pages 70–75, 2004.

[11] Shaodian Zhang, Nóemie Elhadad, Unsupervised Biomedical Named Entity Recognition: Experiments with Clinical and Biological Texts, J Biomed Inform. 2013.

[12] Chen Y, Lasko TA, Mei Q, Denny JC, Xu H. A Study of Active Learning Methods for Named Entity Recognition in Clinical Text. Journal of biomedical informatics. 58:11-18. 2015.

[13] K. Gojenola, M.Oronoz, A. Pérez, A. Casillas. IxaMed: Applying Freeling and a Perceptron Sequential Tagger at the Shared Task on Analyzing Clinical Texts", Proceedings of the 8th International Workshop on Semantic Evaluation , pages 361–365, Dublin, Ireland, August 23-24, 2014.

[14] Fernando Aparicio et al. TMT: A tool to guide users in finding information on clinical texts. Procesamiento del Lenguaje Natural, [S.l.], v. 46, p. 27-34, feb. 2010.

[15] Katona, Melinda and RichárdFarkas. "SZTE-NLP: Clinical Text Analysis with Named Entity Recognition." SemEval@COLING (2014).

[16] Tseytlin E, Mitchell K, Legowski E, Corrigan J, Chavan G, Jacobson RS. NOBLE - Flexible concept recognition for large-scale biomedical natural language processing. BMC Bioinformatics. 2016.

[17] SNOMED® international delivers SNOMED clinical terms®. 2017.

[18] SNOMED CT® Starter Guide - International Release. 2017.

[19] Mikolov, Tomas, Chen, Kai, Corrado, Greg, and Dean, Jeffrey, Efficient estimation of word representations in vector space. 2013a

[20] Y. Bengio, R. Ducharme, P. Vincent. A neural probabilistic language model. Journal of Machine LearningResearch, 3:1137-1155, 2003.

[21] Pastor, Mª Dolores, Navalon, Rafael, Manual de Codificacion CIE-10-Diagnosticos, ministerio de sanidad. 2016.

[22] L.J.P. van der Maaten and G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9(Nov):2579-2605, 2008.

[23] G.E. Hinton and S.T. Roweis. Stochastic Neighbor Embedding. In Advances in Neural Information Processing Systems, volume 15, pages 833–840, Cambridge, MA, USA, 2002.