Semantic Tag Medical Concept using Word2Vec representation

Ignacio Martínez Soriano

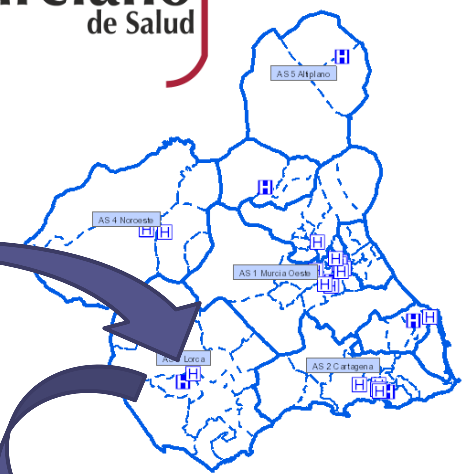
AREA III
Hospital Universitario
"Rafael Méndez"
MEDLAB
MEDIA GROUP

Juan Luis Castro Peña

UNIVERSIDAD
DE GRANADA

Where we are?

Servicio Murciano de Salud



AREA III

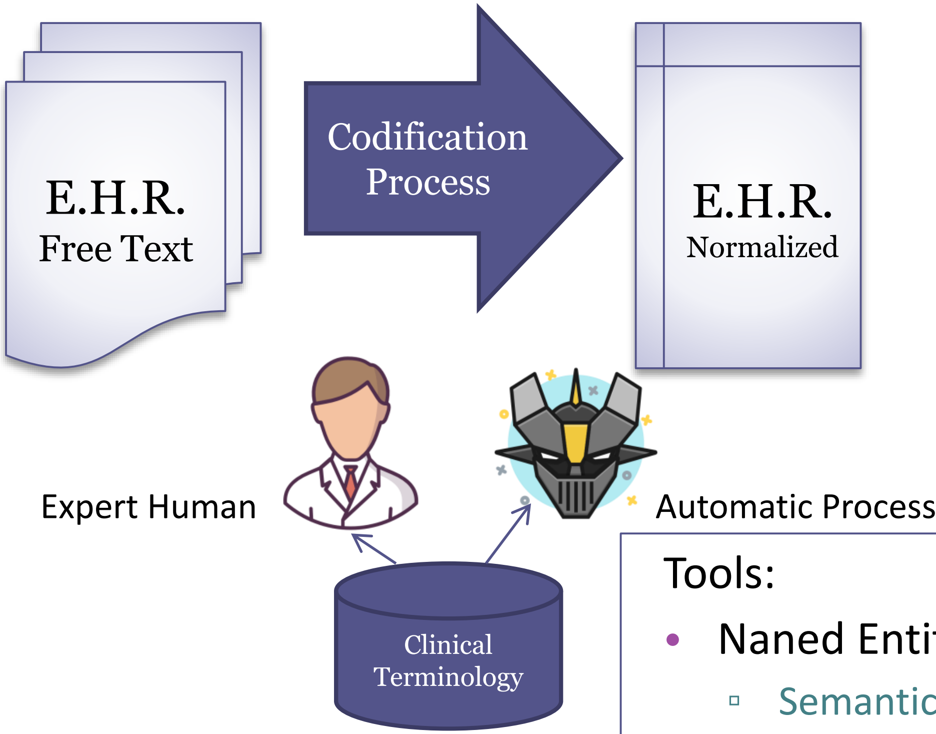
Hospital Universitary “Rafael Méndez”

Lorca (Murcia)

Final Goal: (Semantic Search E.H.R)

To develop a Semantic Search engine, we need that the clinical Information, from free text data, it'll be map with a clinical terminology, like (Snomed-CT, ICD-10-MC, UMLS, etc)

Our Goal: (Semantic Tag Clinical Concepts)



Tools:

- Naned Entity Recognition, (NER).
  - Semantic Tag

Background and Related Work:

Semantic Tag: It is a process of associating an element from a ontology with some document.

S. T. Medical: It is to map clinical concepts from free text clinical reports with a clinical ontology.

Classical Semantic Tag tools:

MedLEE

MetaMap

KnowledgeMap

cTAKES

Supervised machine learning methods like CRF (Condition Random Fields), SSVM (Structural support Vector Machines), and UMLS MetaThesaurus, like Clinical Terminology.

Our approach is to use an unsupervised M.L. Neural Network to discover Word Embedding (Word2Vec) with algorithm rules and Snomed-CT like clinical terminology

Semantic Tag Medical Concepts (STMC):

- We proposed a **mapping tool** to discover from **free text** to **clinical concepts** using the ontology **clinical terminology**, Snomed-CT.
- We use word embedding model (**Word2Vec**) to represents the word in the texts by **vectors** and identify the semantic relation between there.
- We use Named-Based techniques combined with a **query expansion system**, and the Space vector Model, generate with Word2Vec, to find **alternative search terms**.

What is Word Embedding?

- In Spanish there is a proverb:

"Dime con quien andas y te diré quien eres". [El Quijote II 10 y 23].

"Tell me who are your friends and I'll tell you who you are".

To identify the semantic meaning of a word, it depend of the words around it.

What is Word2Vec?

Created by Tomas Mikolov et al. at Google.

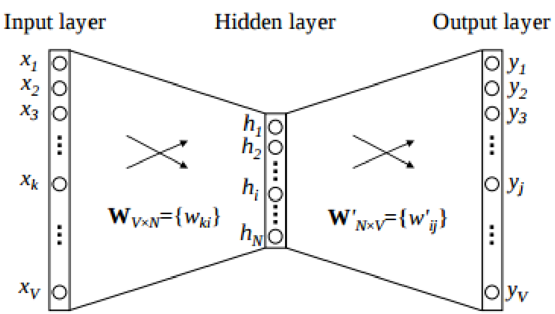
Word2vec is a group of related models that are used to produce word embeddings.

These models are **shallow, two-layer neural networks** that are trained to reconstruct **linguistic contexts of words**.

Word2vec takes as its input a **large corpus** of text and produces a **vector space**,with hundred dimensions each unique word in the corpus.

Characteristics Word2Vec – Structure:

The neural network structure of word2vec is a feedforward network with **one hidden** layer.



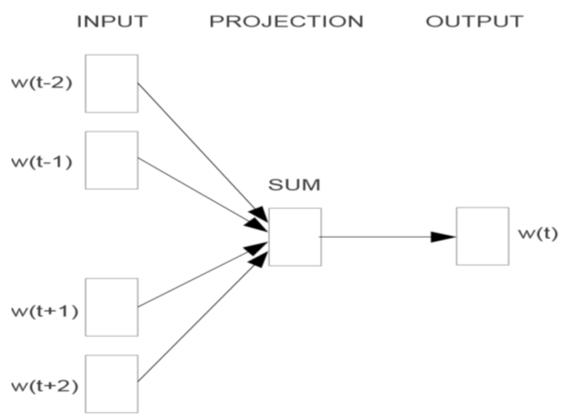
SoftMax Function:

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^{\top} v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left({v'_{w}}^{\top} v_{w_I}\right)}$$

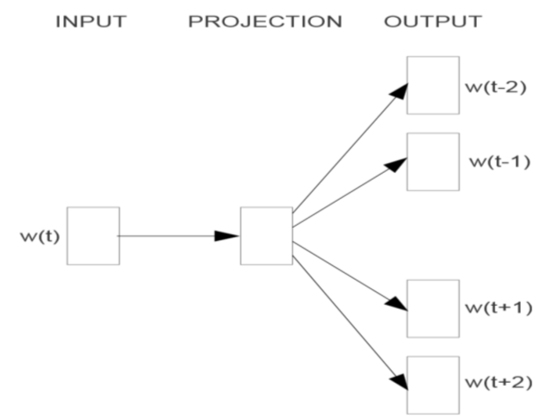
Training can be made feasible by using either ***hierarchical softmax*** or ***negative sampling*** (Mikolov et al.).

The **training** method of word2vec is backpropagation with **stochastic gradient descent**.

Two Models:
CBOW(Continous Bag of Word)
Skip-Gram.



CBOW



Skip-gram

Word2Vec Skip-Gram Model (1/3):

McCormick, C. (2016, April 19). Word2Vec Tutorial - The Skip-Gram Model. Retrieved from http://www.mccormickml.com

Word2Vec uses a trick:

Param(00): SG= Size (0 –CBOW, 1-Skip-Gram)

We don't train a simple neural network with a single hidden layer to perform a certain task, the goal is just to **learn the weights** of the **hidden layer**.

These **weights** are the "**word vectors**" that we're trying to learn.

Param(01): Size of Vector = Size Hidden layer

Given a specific word (the input word). The network is going to tell us the **probability for every word** in our vocabulary (SoftMax function) of being the "nearby word" that we chose.

Param(02): Nearby word = Window size

Word2Vec Skip-Gram Model (2/3):

McCormick, C. (2016, April 19). Word2Vec Tutorial - The Skip-Gram Model. Retrieved from http://www.mccormickml.com

Model Details:

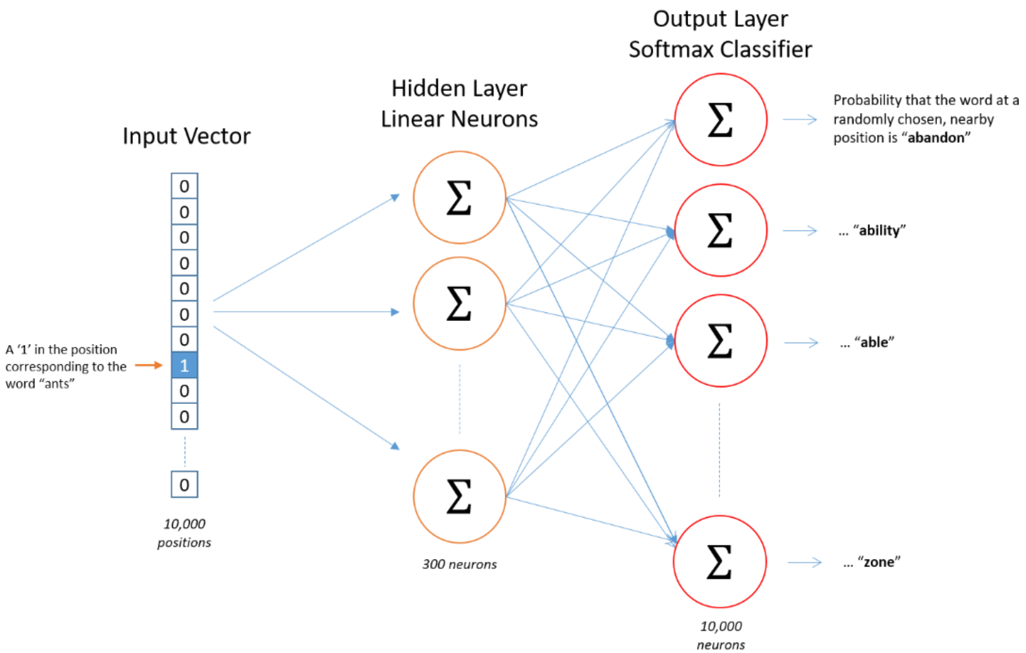
W2V build a vocabulary of words from our training documents.
We have a **vocabulary of 98,103** unique words.

Param(03): min_count(n)= Ignore frequency(w) < n

We're going to represent an **input word** like "fracture" as a **one-hot vector**.

Architecture of our neural network:

The output of the network is a single vector (98103) containing, for every word in our vocabulary, the probability that a randomly selected nearby word is that vocabulary word.



Word2Vec Skip-Gram Model (3/3):

McCormick, C. (2016, April 19). Word2Vec Tutorial - The Skip-Gram Model. Retrieved from http://www.mccormickml.com

There is no activation function on the hidden layer neurons, but the output neurons use **softmax** like clasification method to build a probabilty distribution.

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^\top v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left({v'_w}^\top v_{w_I}\right)}$$

Training de model:

Input: **one-hot** vector for a word

Output: **probability distribution** vector

Param(04): hs= (1 hierarchical softmax, 0 negative sampling)

We're learning word vectors with 300 features. So the hidden layer is going to be represented by a weight matrix with 98103 rows (one for every word in our vocabulary) and 300 columns (one for every hidden neuron)

We use gensim python library (Parameters):

Param(00): SG= 1

Param(01): Size=300

Param(02): window= 5

Param(03): min_count(n)= 2

Param(04): hs= 0

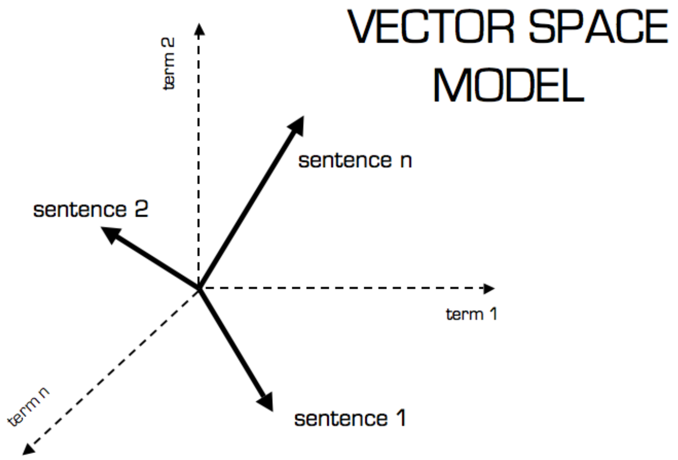
Param(05): negative_sampling= 5

How identify the meaning between two words?

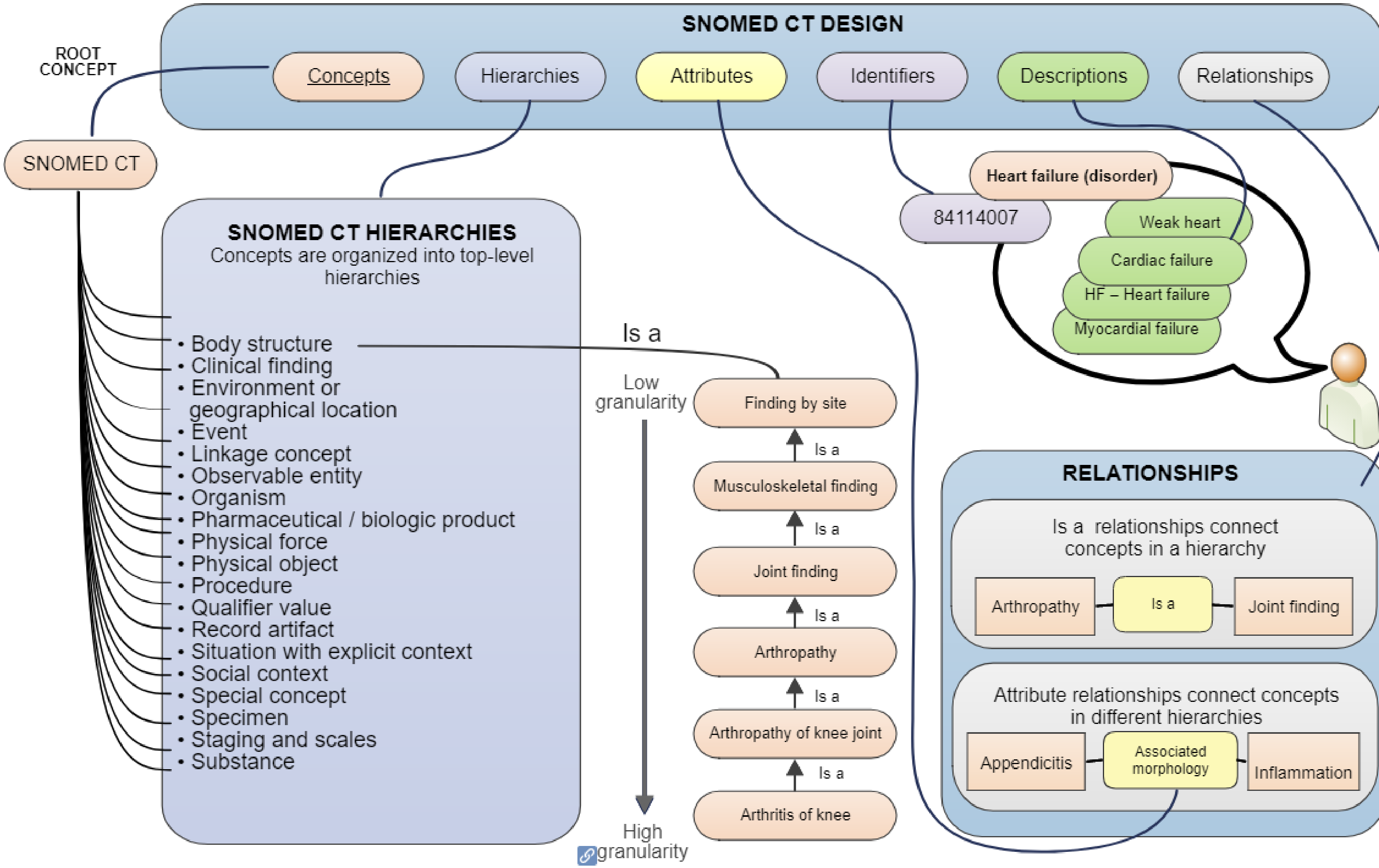
Similarity Distance between Words: Cosine Distance Word Vectors.

Cosine Distance:

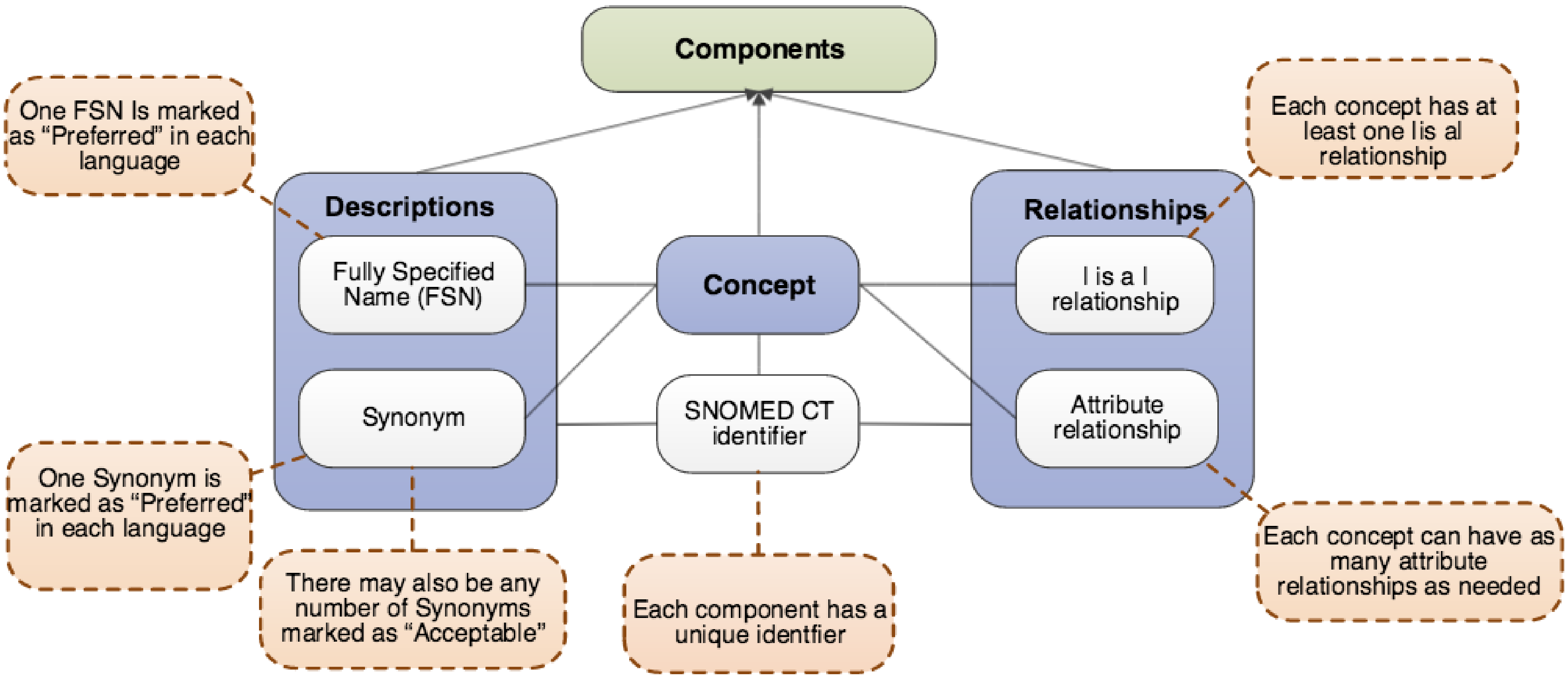
$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$



Snomed-CT Design and Structure

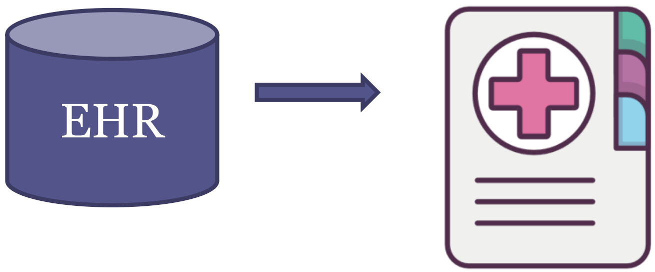


Snomed-CT Components. Logical Model



Implementation S.T.M.C.:

Big Data Corpus:



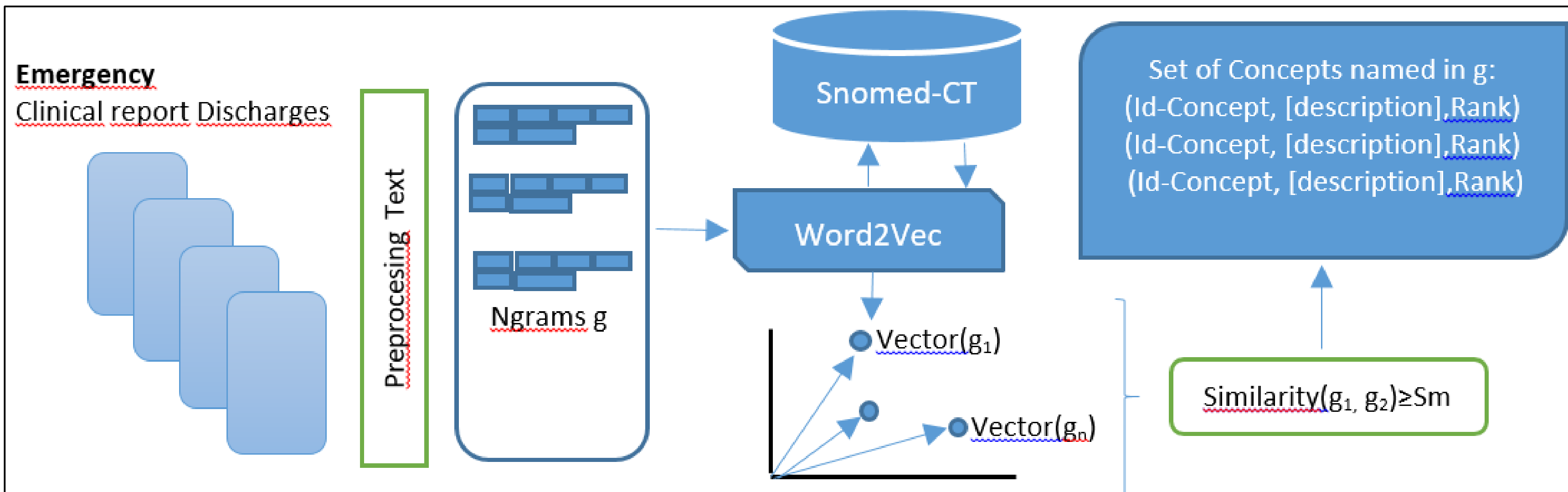
We use 615,513 emergency discharge reports.

Emergency Discharge Records

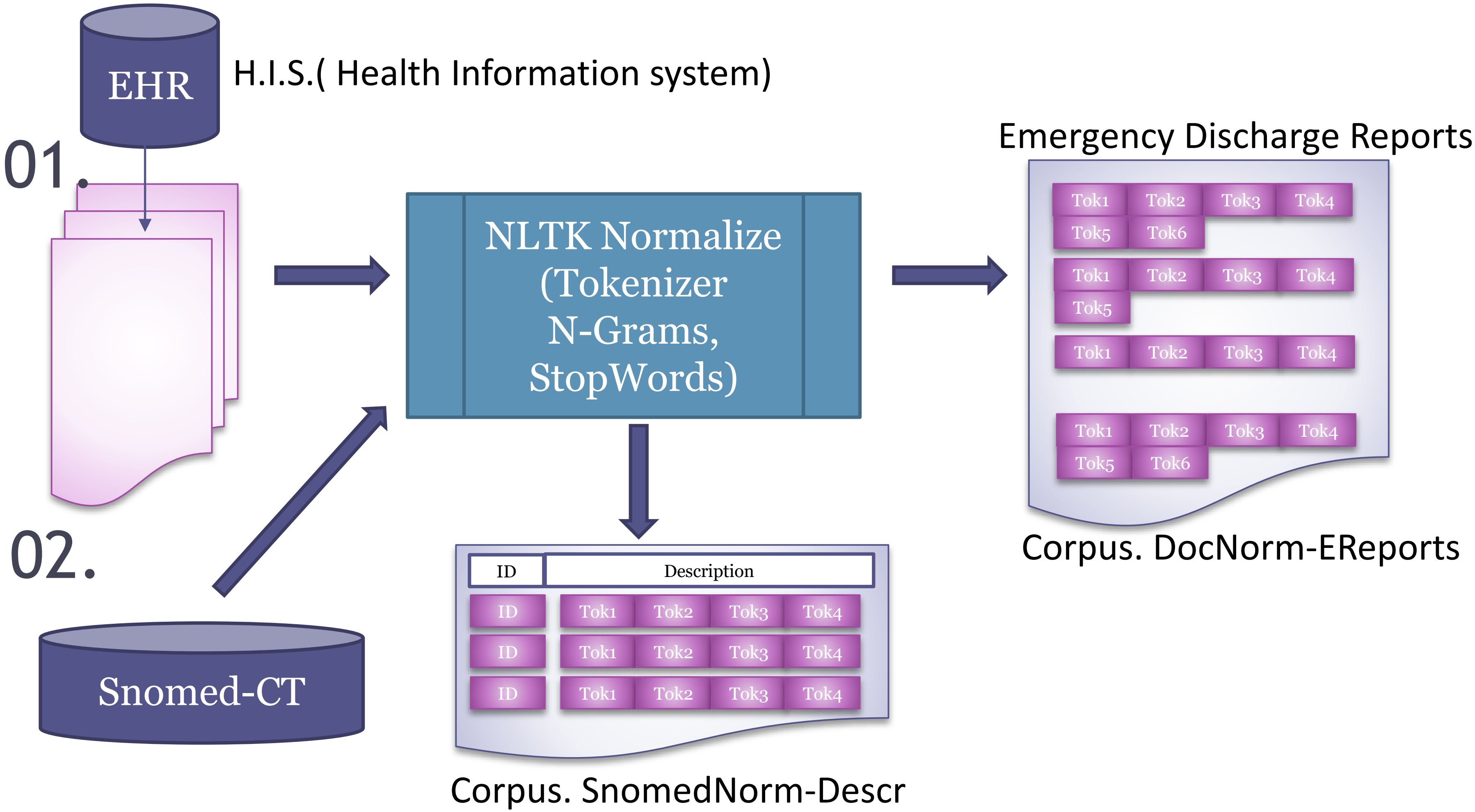
Emergency Electronic Health Records:

- ***Emergency Discharge report:***
  - Administrative Data (anonymised)
  - Reason Medical Consultation
  - Personal Background:
    - Known Allergies.
    - Medicals
    - Surgeries.
    - Treatment Background.
  - Actual illness.
  - Exploration.
  - Complementary Evidence.
  - Evolution.
  - Diagnostic.
  - Treatment and Recommendations.

General Process Diagram:

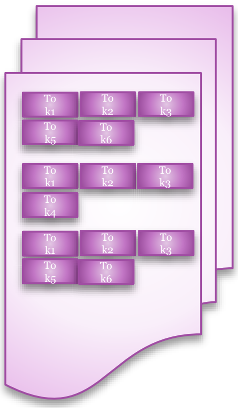


Algorithm STMC: Preprocessing Text



04. Algorithm STMC: Word2Vec Model

Corpus. DocNorm-EReports

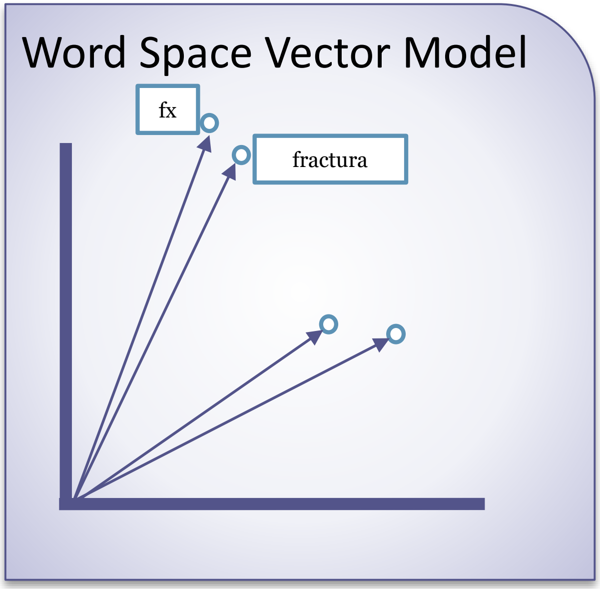


615513 Reports

Gensim Word2Vec

Skip-Gram sg:1
Size: 300
Window:5
Min_count:2
Hs=0

Local Vector Model Domain



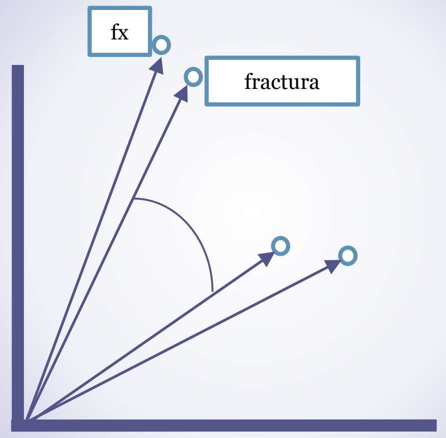
04. Algorithm STMC: Word2Vec Cosine distance

Similarity: Cosine distance

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Distance Cosine

Word Space Vector Model



Model Word2Vec
W2V

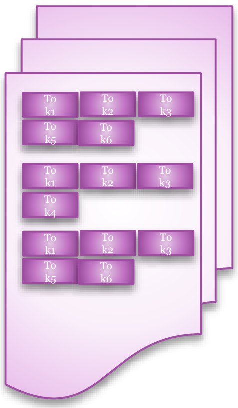
Method:
W2V.most_similar(w)

Function:
Dist_Cosine(w1,w2)

05. Algorithm STMC: (Process)

Step 01

Corpus. DocNorm-EReports



N-Gram

615513 Reports

Step 02

For Every Sentence of the Corpus we get a combination of 3-grams

i.e: Sentence = [‘Acute’,’myocardial’,’infarction’]

1-grams = [[‘Acute’],[’myocardial’],[’infarction’]]

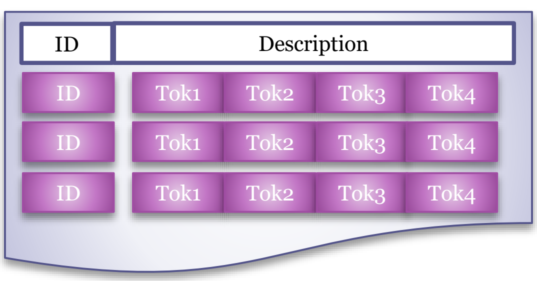
2-grams = [[‘Acute’,’myocardial’],[’myocardial’,’infarction’]]

3-grams = [[‘Acute’,’myocardial’, ’infarction’]]

Step 03

Classical query of Tokens

Corpus. SnomedNorm-Descr



Description
Select Token

Set_Snomed= Set of Possible Candidate Concept

06. Algorithm STMC: (Process)

Step 04

It will denote ***v(w)***, the vector of the word ***w*** ib the Model ***M***

Given a n-gram $\boldsymbol{g} = \boldsymbol{w_1\, w_2 \ldots w_n}$

Given a n-gram $\boldsymbol{v(g)} = \boldsymbol{v(w_1) + v(w_2) + \ldots + v(w_n)}$

We define Similarity between 2 n-gram $\boldsymbol{g_1, g_2}$:

$$Sim() = CosineSimilarity\ (v(g_1), v(g_2))$$

We use the ***similarity*** between ***n-grams*** to identify if a **concept** is named in a sentence ***S***.

Given a concept ***c*** in a ontology ***O***:

Degree in witch a n-gram ***g*** names the concept ***c*** as the maximum similarity between ***g*** and one of the label of ***c***:

$$Names(g,c) = Max\{Sim(g,l);\ l\ label\ of\ c\}$$

Degree in witch ***c*** is named in a sentence ***S***:

$$Names(g,S) = Max\{Names(g,c);\ g\ n\text{-}gram\ of\ c\}$$

If ***Names(c,S)=1***, then ***c*** is standardized named in ***S***

07. Algorithm STMC: (Process)

Step 05: Algorithm two passes

Pass 1: We filter n-gram and concepts to possible candidates in standard way.

For every n-gram ***g*** of **S** and every concept ***c*** of ***O:***

- If ***Names(g,c)=1***, then ***c*** is named in ***S*** by expression ***g.***
- If ***Names(g,c)>alpha(0.9)***, then ***g*** is added to the list og ***GC*** (Grams Candidates), and ***c*** to the list of ***CC***(Concepts Candidates) to be named in ***S.***

Pass 2: We check if some n-gram candidate names a concept in a non standard way.

For every n-gram ***g*** of ***GC***, we get a set of ***variants*** of ***g.***
This variants are generated from ***g*** by replacing some words of ***g*** by one of a list of 5
g'=Most_Similar(w',5)
For every ***variants g'*** of a n-gram ***g*** of ***G***, and any concept ***c*** of ***CC.***

- ***If Names(g',c)=1,*** then it is identified that ***c*** is named in ***S*** by the expression ***g.***

08. Bag of Clinical Concepts (BOCC):

We can represents a Medical reports as a **bag of concepts,** similar way like bag of Words.

Given a concept ***c*** and a document ***d***, we define the ***frequency of a concept*** in the document:

CF(c,d) = |g in d; c named by the expression g|

We represent a document ***d***, by the frequency of concepts of ***O*** in ***d:***

CF(d) = {(c, CF(c,d)); c in O}

Or simplifying, Concept Frequency reduced Representation:

- If ***c*** is not named in ***d,*** then ***c*** is not considered in the reduced representation
- if c_1 and c_2 are named in ***d,*** and c_1 is more detailed than c_2 in the ontology hierarchy, then we say that c_2 is subsumed by c_1. ***Subsumed***(c_2,c_1), and not considered in reduced representations.

C(d) = {c in O; CF(c,d) > 0}

$$\mathrm{MaxC(d)}=\{\boldsymbol{c} \in \boldsymbol{C}(\boldsymbol{d});\ \forall \boldsymbol{c}' \in \boldsymbol{C}(\boldsymbol{d})\ \ \neg\, \boldsymbol{Subsumed}(\boldsymbol{c},\boldsymbol{c}')\}$$

Then we have the reduced representation by:

$$\boldsymbol{CFR}(\boldsymbol{d}) = \{(\boldsymbol{c}, \boldsymbol{CF}(\boldsymbol{c},\boldsymbol{d})); \boldsymbol{c}\ \boldsymbol{in}\ \boldsymbol{MaxC}(\boldsymbol{d})\}$$

09. Uses Cases Examples:

Example: How Algorithm identify a conceptID named In no normalized way

In [20]:
```
CN=[]
GCN=[]
CC=[]
GC=[]

CN, GCN, CC, GC = paso1(Sent5, 0.9, 300)
```

```
Paso.1:
----------------------
01. Crear ngramas:[['rodilla'], ['dcha'], ['rodilla', 'dcha']]
----------------------------------------------------------
03. Total conceptos de snomed-CT donde aparece alguna etiqueta "l": 1073
----------------------------------------------------------
05. (CN) -> Conceptos Nombrados, umbral alfa(1.0): [(1.0, ['rodilla'], ['rodilla'], '72696002')]
----------------------------------------------------------
06. (GCN) -> Gramas Candidatas Nombradas, umbral alfa(1.0): [['rodilla']]
----------------------------------------------------------
07. (GC) -> Gramas Candidatas, umbral alfa(0.9):
id: 0 - ['rodilla', 'derecha']
----------------------------------------------------------
08. (CC) -> Conceptos Candidatos, superan el umbral alfa(0.9):
id: 0 - (0.9056999999999995, ['rodilla', 'dcha'], ['rodilla', 'derecha'], '6757004')
id: 1 - (0.9056999999999995, ['rodilla', 'dcha'], ['rodilla', 'derecha'], '210562007')
id: 2 - (0.9056999999999995, ['rodilla', 'dcha'], ['rodilla', 'derecha'], '210562007')
----------------------------------------------------------
```

10. Uses Cases Examples:

<u>Example:</u> Identify, relations and similar words, to discover new words meaning

```
In [17]: InfAltModel.most_similar(['fx'])

Out[17]: [('fractura', 0.8746222853660583),
          ('fract', 0.7706700563430786),
          ('fratura', 0.7633858919143677),
          ('frx', 0.7520125508308411),
          ('frac', 0.7503272294998169),
          ('fisrua', 0.7457991242408752),
          ('desplazada', 0.7253227233886719),
          ('conminuta', 0.7239192724227905),
          ('diafisis', 0.7234492897987366),
          ('fracrtura', 0.7221113443374634)]
```

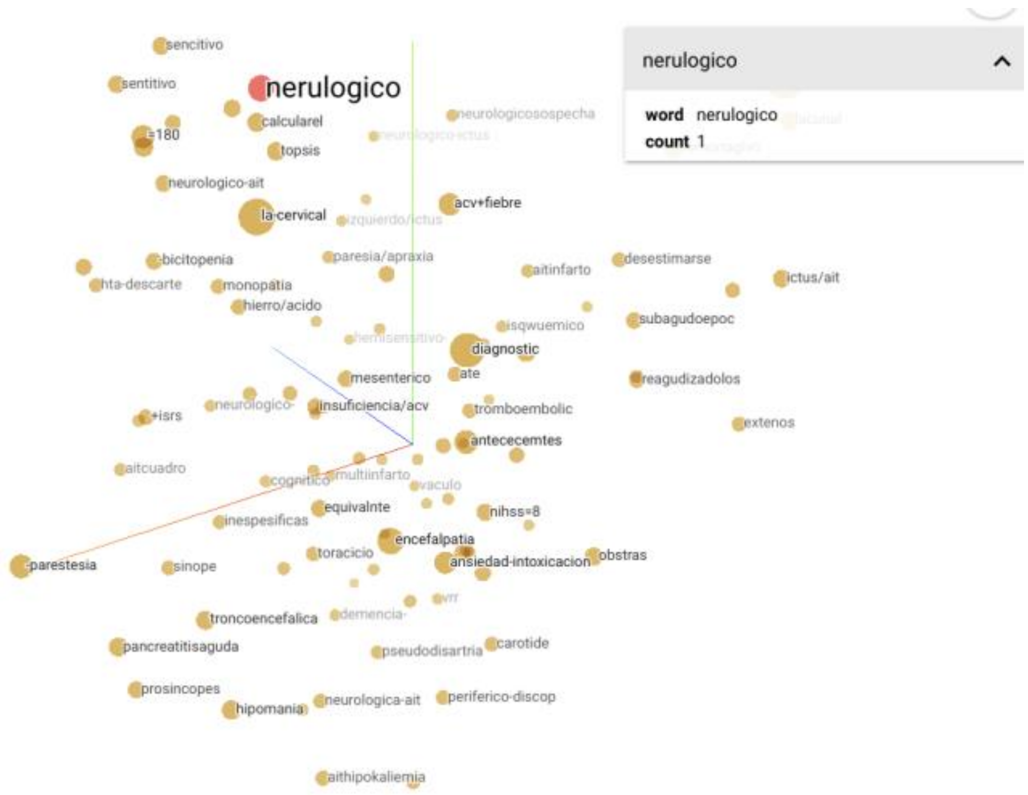
```
In [20]: InfAltModel.most_similar(['dcha'])

Out[20]: [('izda', 0.8290866613388062),
          ('derecha', 0.7282750606536865),
          ('izqda', 0.7253247499465942),
          ('dercha', 0.7189282178878784),
          ('decha', 0.7020336389541626),
          ('dcho', 0.6982229948043823),
          ('dch', 0.680919349193573),
          ('izquierda', 0.6796815395355225),
          ('izd', 0.673693180084285),
          ('deercha', 0.666525661945343)]
```

```
In [15]: InfAltModel.most_similar(['tce'])

Out[15]: [('tec', 0.7197970151901245),
          ('craneoencefalico', 0.6779731512069702),
          ('conocimiento', 0.6762552857398987),
          ('conmocion', 0.6442399621009827),
          ('politraumatismo', 0.6233476400375366),
          ('conocimento', 0.6212176084518433),
          ('occipital', 0.6101372838020325),
          ('p/c', 0.5979176759719849),
          ('pdc', 0.584796667098999),
          ('policontusionado', 0.5799985527992249)]
```

```
In [16]: InfAltModel.most_similar(['hta'])

Out[16]: [('hipertensiva', 0.7690305113792419),
          ('hipertension', 0.6535224914550781),
          ('captopril', 0.6313796043395996),
          ('hipertesiva', 0.6241427659988403),
          ('htva', 0.6211124658584595),
          ('crisisi', 0.6208800077438354),
          ('anisedad', 0.6053137779235840),
          ('capoten', 0.6037868857383728),
          ('resuelta', 0.596153199672699),
          ('hta-', 0.5947197675704956)]
```

11. Uses Cases Examples:

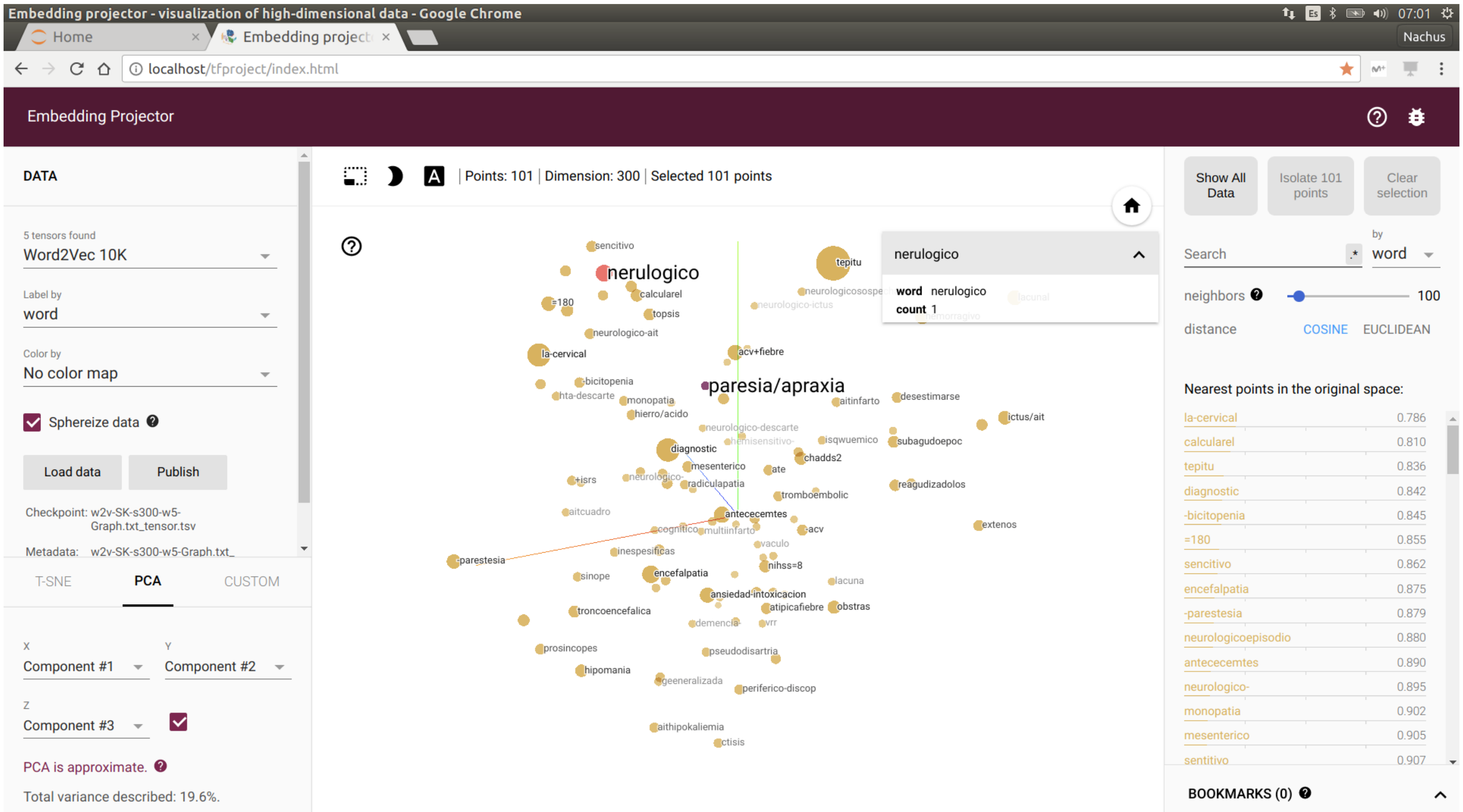
<u>Example:</u> Identify, relations and similar words, to discover new words meaning




```
In [18]: InfAltModel.most_similar(['od'])
Out[18]: [('oi', 0.9486678838729858),
 ('ojo', 0.6543353199958801),
 ('aos', 0.6343369483947754),
 ('corneal', 0.6270889043807983),
 ('tarsal', 0.6240204572677612),
 ('oizq', 0.6226646304130554),
 ('oj', 0.6169692277908325),
 ('entropion', 0.6134785413742065),
 ('catarata', 0.6108546853065491),
 ('queratitis', 0.6084508895874023)]

In [19]: InfAltModel.most_similar(['oi'])
Out[19]: [('od', 0.9486678838729858),
 ('ojo', 0.6585066318511963),
 ('oizq', 0.6367001533508301),
 ('tarsal', 0.6354893445968628),
 ('aos', 0.6344435214996338),
 ('queratitis', 0.6298179626464844),
 ('corneal', 0.6265361309051514),
 ('oj', 0.6230565309524536),
 ('entropion', 0.6230305433273315),
 ('uveitis', 0.6224710941314697)]
```

11. Uses Cases Examples:

Example: Visualization Examples(Tensorflow project):



Evaluation:

Corpus Gold:
we generate a Corpus gold from the Emergency discharge clinical reports with the help of two Expert, using the Browser ihtsdotools to codify the reports. (http://browser.ihtsdotools.org/?)

We use Precision, recall and F_Measure to analyze the performance tool

Precision: P=TP/(TP+FP)

Recall: R=TP/(TP+FN)

$$F\_Measure = (1+\beta^2) * \frac{precision * recall}{(\beta^2 * precision) + recall}$$

$\beta = 0.7$ To put more emphasis on precision than recall

TABLE I. TABLE MEASURES		
Concept	Our approach	
Precision	0.8097	
Recall	0.7469	
F-Measure	0.7879	

Conclusion and Outlook

This technology can use in many practical applications:

Future applications to develop:

- Semantic Search from free text Electronic Health Records.
- A tool assistant, to help the human expert, to assign the correct clinical id concept, from clinical reports.
- Discover new local words from a closed clinical domain.
- Identify and disambiguate abbreviations from a local clinical domain.
- Identify relations between type mistakes and the correct word.
- A new kind of visualization concept, using the vector Space Model.

Tack så mycket

Acknowledgment

- To my Director PH.D. Thesis:
  - D. Juan Luis Castro Peña.
- Medlab Mediagroup.
- Hospital Universitary “Rafael Méndez”