

## **Word embedding for French natural language in healthcare: a comparative study**

Emeric Dynamant, Romain Lelong, Badisse Dahamna, Clément Massonau, Gaétan Kerdelhué, Julien Grosjean, Stéphane Canu, Stefan J Darmoni

Submitted to: JMIR Medical Informatics  
on: September 25, 2018

**Disclaimer:** © The authors. All rights reserved. This is a privileged document currently under peer-review/community review. Authors have provided JMIR Publications with an exclusive license to publish this preprint on its website for review purposes only. While the final peer-reviewed paper may be licensed under a CC BY license on publication, at this stage authors and publisher expressly prohibit redistribution of this draft paper other than for review purposes.

## Table of Contents

---

Original Manuscript .....	4
Supplementary Files.....	28
Figures.....	29
Figure 4.....	30



# Word embedding for French natural language in healthcare: a comparative study

Emeric Dynamant<sup>1,2,3</sup>, M.Sc.; Romain Lelong<sup>2,3</sup>, MSc; Badisse Dahamna<sup>2,4</sup>, Ph.D.; Clément Massonau<sup>2</sup>, M.D.; Gaétan Kerdelhué<sup>2,4</sup>, M.Sc.; Julien Grosjean<sup>2,4</sup>, PhD; Stéphane Canu<sup>3</sup>, Ph.D.; Stefan J Darmoni<sup>2,4</sup>, M.D., Ph.D.

## Corresponding Author:

Emeric Dynamant, M.Sc.

OmicX

72 Rue de la République

Le Petit Quevilly

France

Phone: 33 659901249

Email: [emeric.dynamant@omictools.com](mailto:emeric.dynamant@omictools.com)

## Abstract

**Background:** Word embedding technologies are now used in a wide range of applications. However, no formal evaluation and comparison have been made on models produced by the three most famous implementations (Word2Vec, GloVe and FastText).

**Objective:** The goal of this study is to compare embedding implementations on a corpus of documents produced in a working context, by health professionals.

**Methods:** Models have been trained on documents coming from the Rouen university hospital. This data is not structured and cover a wide range of documents produced in a clinic (discharge summary, prescriptions ...). Four evaluation tasks have been defined (cosine similarity, odd one, mathematical operations and human formal evaluation) and applied on each model.

**Results:** Word2Vec had the highest score for three of the four tasks (mathematical operations, odd one similarity and human validation), particularly regarding the Skip-Gram architecture.

**Conclusions:** Even if this implementation had the best rate, each model has its own qualities and defects, like the training time which is very short for GloVe or morphosyntactic similarity conservation observed with FastText. Models and test sets produced by this study will be the first publicly available through a graphical interface to help advance French biomedical research.

(JMIR Preprints 25/09/2018:12310) DOI: <https://doi.org/10.2196/preprints.12310>

## Preprint Settings

1) Would you like to publish your submitted manuscript as preprint?

✓ (a) Yes, please make my preprint PDF available to anyone at any time (Recommended).

(b) Yes, but please make my preprint PDF available only to logged-in users; I understand that my title and abstract will remain visible to

(c) Yes, but only make the title and abstract visible.

(d) No, I do not wish to publish my submitted manuscript as a preprint.

2) If accepted for publication in a JMIR journal, would you like the PDF to be visible to the public?

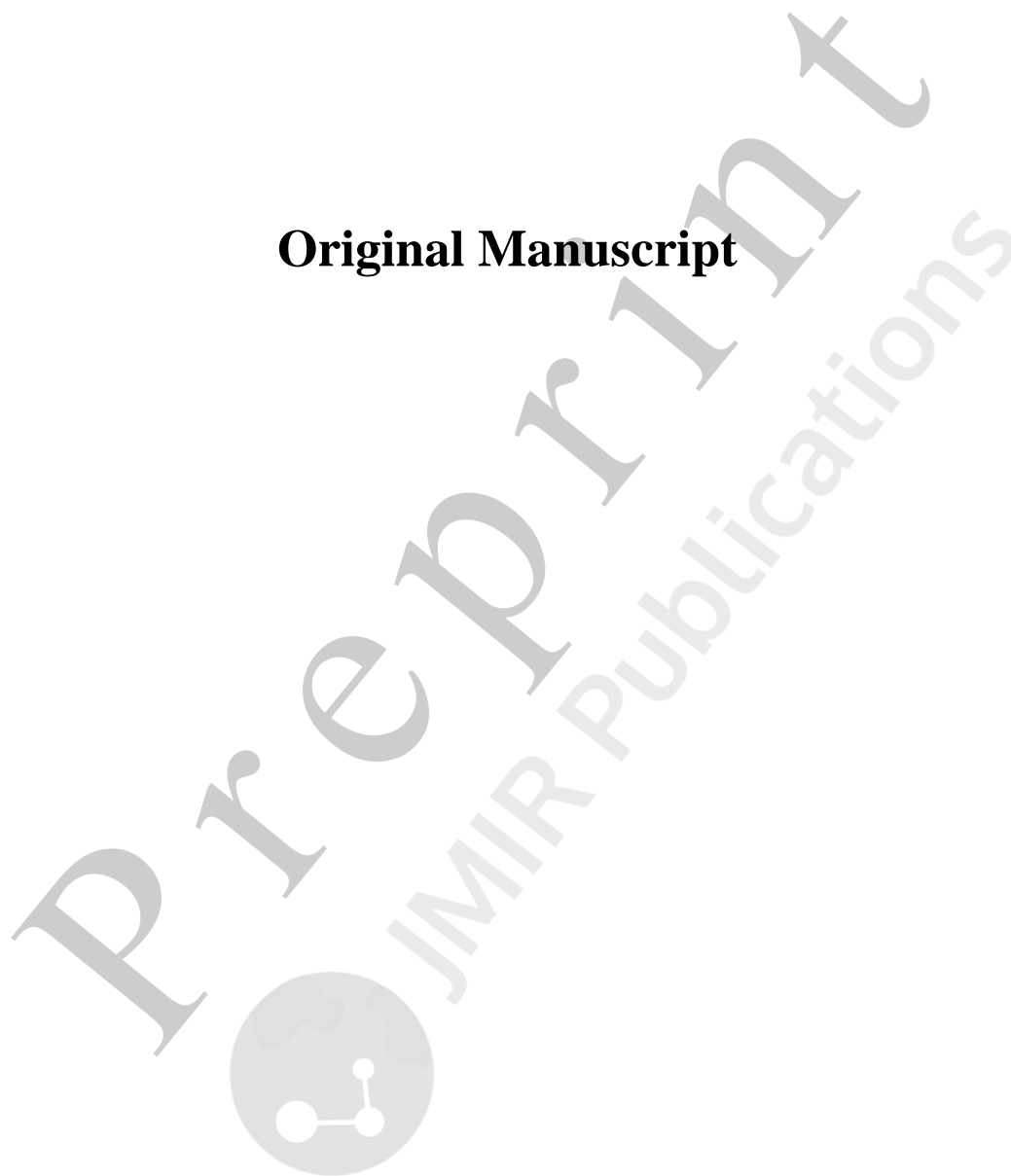
✓ (a) Yes, please make my accepted manuscript PDF available to anyone at any time (Recommended).

(b) Yes, but please make my accepted manuscript PDF available only to logged-in users; I understand that the title and abstract will remain

(c) Yes, but only make the title and abstract visible (see Important note, above). I understand that if I later pay to participate in <a href="h

**Original Manuscript**

Preprint  
JMIR Publications



Emeric DYNAMANT<sup>1,2,3,\*</sup>, Romain LELONG<sup>2,3</sup>, Badisse DAHAMNA<sup>2,4</sup>, Clément MASSONNAUD<sup>2</sup>, Gaétan KERDELHUÉ<sup>2,4</sup>, Julien GROSJEAN<sup>2,4</sup>, Stéphane CANU<sup>3</sup>, and Stéfan DARMONI<sup>2,4</sup>

<sup>1</sup>OmicX, 72 Rue de la République, 76140, Le Petit Quevilly, Normandie, France

<sup>2</sup>Department of Biomedical Informatics, Cour Leschevin, CHU de Rouen, 1 Rue de Germont, 76031 Rouen, Normandie, France

<sup>3</sup>LITIS, Université de Rouen Normandie, UFR Sciences et Techniques, Avenue de l'Université, 76800, Saint-Étienne-du-Rouvray, Normandie, France

<sup>4</sup>LIMICS, Campus des Cordeliers, INSERM U1142, 15 Rue de l'École de Médecine, 75006, Paris, France

\* Corresponding author: [emeric.dynamant@omictools.com](mailto:emeric.dynamant@omictools.com), +33232888829, CISMef, Department of Biomedical Informatics, Cour Leschevin, CHU de Rouen, 1 Rue de Germont, 76031 Rouen, Normandie, France.

## Word embedding for French natural language in healthcare: a comparative study

### Abstract

**Background:** Word embedding technologies are now used in a wide range of applications. However, no formal evaluation and comparison have been made on the ability of each of the three current most famous unsupervised implementations (Word2Vec, GloVe and FastText) to keep track of the semantic similarities existing between words, when trained on the same dataset.

**Objectives:** The goal of this study was to compare embedding methods trained on a corpus of French health-related documents produced in a professional context. The best method will then help us to develop a new semantic annotator.

**Methods:** Unsupervised embedding models have been trained on than 641,279 documents originating from the Rouen university hospital. These data are not structured and cover a wide range of documents produced in a clinical setting (discharge summary, procedure reports, prescriptions, etc). Four rated evaluation tasks were defined (cosine similarity, odd one, analogy-based operations and human formal evaluation) and applied on each model, as well as embedding visualization.

**Results:** Word2Vec had the highest score on three out of four rated tasks (analogy-based operations, odd one similarity and human validation), particularly regarding the Skip-Gram architecture.

**Conclusions:** Although this implementation had the best rate for semantic properties conservation, each model has its own qualities and defects, like the training time which is very short for GloVe or morphological similarity conservation observed with FastText. Models and test sets produced by this study will be the first publicly available through a graphical interface to help advance French biomedical research.

**Keywords:** Natural Language Processing (D009323); Data Mining (D057225); Data Curation (D066289)

### Introduction

#### Context

The use of clinically derived data from Electronic Health Records (EHRs) and other clinical information systems can greatly facilitate clinical research as well as optimizing diagnosis related groups or other initiatives. The main approach for making such data available is to incorporate them from different sources into a joint Health Data Warehouse (HDW), thus containing different kinds of

natural language documents such as prescription, letters, surgery reports... all written in everyday language (spelling errors, acronyms, short and incomplete sentences, etc).

Clinical Named Entity Recognition (NER) is a critical Natural Language Processing (NLP) task to extract concepts from named entities found in clinical and health documents (including discharge summaries). A Semantic Health Data WareHouse (SHDW) was developed by the Department of Biomedical Informatics of the Rouen University Hospital (RUH), Normandy, France. It is composed of three independent layers based on a NoSQL architecture:

- A cross-lingual terminology server, HeTOP, which contains 75 terminologies and ontologies in 32 languages [1].
- A semantic annotator based on NLP bag-of-words methods (ECMT) [2].
- A semantic multilingual search engine [3].

In order to improve the semantic annotator, it is possible to implement deep learning techniques to the already existent one. To do so, a new text representation, which keeps the most semantic similarities existing between words, has to be designed to fit the input of neural networks algorithms (text embedding).

## Word embedding

In NLP, finding a text representation which retains the meaning proximities has always been a moot point. Indeed, the chosen representation has to keep the semantic similarities between different words from a corpus of texts in order to allow indexation methods to output a correct annotation. Thus, the representation of a unique token has to show the proximity with other related meaning concepts (synonyms, hyponyms/cohyponyms, other related tokens, etc), as illustrated in the quotation “*You shall know a word by the company it keeps*” [4], now known as the *distributional hypothesis*.

During the sixties, the System for the Mechanical Analysis and Retrieval of Text (SMART) information retrieval system brought the Vector Space Model (VSM), which led to the idea of vectorial representation of words [5, 6]. With this approach, the word vectors were sparse (the encoding of a word being a vector of  $n$  dimensions,  $n$  representing the vocabulary size). In fact, a compact and precise representation of words could bring several benefits. First comes the computational aspect. Computers are way better to perform operations on low-dimensional objects. This then permits to calculate the probability of a specific concept to appear close to another one. Moreover, the vectors' dimensions created to represent a word can be used to fit this word in a space and thus make distance comparisons with other tokens. Current unsupervised embeddings techniques provide dense and low-dimensional information about a word, either with count-based or predictive-based methods [7]. Different implementations of techniques mapping words into a VSM have been developed.

## Word2vec

The word2vec approach was the first modern embedding released in 2013 [8]. Mikolov *et al.* implemented two kinds of architectures: the Continuous Bag-Of-Word (CBOW) and the Skip-Gram (SG).

The **CBOW architecture** is learning to predict a target word **W** by using its context **C**. This model is similar to a feedforward neural network proposed before [8, 9]. However, the bias brought by the non-linear layer has been removed with a shared projection layer. The input layer accepts one-hot encoding as input  $X_i$  (a sentence is encoded as a very hollow vector. It is composed of 0 or 1,

depending on the words found in this sentence, and becomes  $\mathbf{X}'_i$  when passing through the activation function). With a corpus composed of  $V$  different words and an input layer size of  $N$  chosen, the hidden representation of this corpus will be a  $V \times N$  matrix with each row representing a word  $\mathbf{W}_v$  by a vector of dimension  $N$ . After passing through the linear activation function of the hidden layer, the output  $\mathbf{Y}_i$  can be computed using the softmax function presented below for each word  $W \in V$  [10].

$$Y_i = \frac{e^{x_i}}{\sum_{j=1}^v e^{x'_j}}$$

**The SG architecture** is using a given word to predict its context, unlike the CBOW architecture. The entire corpus  $V$  will thus be transformed into many couples *target || context* (i.e. *input || output* or  $x_i || y_i$  of the network) and a stochastic gradient descent optimizing function will be used on this training dataset with a minibatch parsing, as defined below [11].

$$loss = - \sum_{c=1}^c x_i + C \cdot \log \left( \sum_{v=1}^v e^{x'_i} \right)$$

Thus, the hidden and the output weights matrix will have a shape of  $V \times N$ , with  $N$  being again the number of dimension for word vectors. To reduce the computation of such an amount of data (in a “normal” training, all the weights of the network should be updated for each passage through an example. The amount of changes depending on the size of the contextual windows), the authors brought some new ideas. First, words pairs appearing always together are treated as a single token for both architectures (“*New York*” is much more meaningful than the combination of “*New*” and “*York*”). Then, as specifically regards the SG architecture, frequent words subsampling (kind of dropout, the network has a chance to reinitialize a word vector to reduce the over-updating of some common words) is applied and the negative sampling make the model to update only a portion of the context for each target [12].

## GloVe

This model is the embedding released by the Stanford University [13]. Like Word2vec, GloVe can embed words as mathematical vectors. However it differs on the method used to capture similarity between words, GloVe being a count-based method. The idea was to construct a huge co-occurrence matrix between the words found in the training corpus of shape  $V \times C$  with  $V$  being the vocabulary of the corpus and  $C$  context examples. The probability  $P(V_{w1} || V_{w2})$  of a word  $V_{w1}$  being close to another  $V_{w2}$  will increase during the training and fill the co-occurrence matrix. This gigantic matrix is then factorized by using the log function, this idea coming from the LSA model [14].

## FastText

It is a newly released model in 2017, which comes from a new idea [15]. While both Word2Vec and GloVe assumed that a word can be effectively and directly embedded as a vector, Bojanowski *et al.* [15] consider that a word could be the result of all of the vectorial decomposition of this word (subword model). Each word  $V_w$  with  $V$  being the vocabulary can be decomposed into a set of  $n$ -characters-grams vectors. For example, the word “*boat*” can be seen as

$\vec{b} + \vec{bo} + \vec{boa} + \vec{o} + \vec{oa} + \vec{oat} + \vec{a} + \vec{at} + \vec{t}$  (with the n-gram parameter  $n = 3$ , indicating the maximum number of letters composing a sub-word). Thus, each word is embedded in the vectorial space as the sum of all vectors composing this token, incorporating morphological information into the representation [16]. Like Word2Vec, FastText comes with the two different architectures (SG and CBOW).

## Related work

Since a few years, the huge interest in words embeddings led to comparison studies. Scheepers *et al.* compared the three word embedding methods but these models were trained on different and non-specific datasets (Word2Vec on news data, while FastText and GloVe trained on more academic data, Wikipedia and Common Crawl respectively, a bias could have been brought by such a difference) [17]. Bairong *et al.* also performed a comparison between these three implementations, but focused on bilingual automatic translation comparison (BLEU score [18]) and without human evaluation for all the different models. The goal here is to determine the best ability to keep semantic relationships between words [19]. More recently, Beam *et al.* produced huge publicly available word embeddings based on medical data, however this work did not involve FastText, only Word2Vec and GloVe. Moreover, the benchmark between embeddings methods was based on statistical occurrences of the concepts [20]. In a similar way, Huang *et al.* deeply studied Word2Vec on three different medical corpuses, measuring the impact of the corpus' focus on medicine and without evaluating the semantic relationships [21]. Finally, Wang *et al.* compared word embeddings training set influence on models used for different NLP tasks related to medical applications, while the goal here is to compare embedding implementations trained on the same corpus [22].

Moreover, many different teams or companies have released pre-trained word embedding models (*e.g.*: Google, Stanford University, etc), which could be used for specific applications. Wang *et al.* also proved that word embeddings trained on highly specific corpus are not so different than those trained on publicly available and general data such as Wikipedia [22]. However, in a clinical context, the vocabulary coverage of those embeddings, trained on academic corpus, are quite low regarding the words used in a professional context. To assess the proportion of these non-overlapping tokens, 1,250,000 articles abstracts were extracted from the French scientific articles database LiSSa and they have been compared to the health raw data from the SHDW [23]. These health documents contained 180,362,939 words in total representing 355,597 unique tokens, and the abstracts from the LiSSa database are composed of 61,119,695 words representing 380,879 unique tokens. Among the 355,597 unique tokens written in the SHDW documents, 92,856 (26.1%) were not found in the abstract from the LiSSa corpus (mainly representing misspells, acronyms or geographic locations). Thus, more than a quarter of the vocabulary used in professional context cannot be better embedded by using an academic pre-training corpus. Thus, a local training on specific data is often needed, especially with languages other than English, with less trained embeddings are available.

## Contributions

Words embedding comparisons thus have previously been studied, but as far as we know, none of them compared the ability of the five actual most used unsupervised embedding implementations trained on a medical dataset produced in a professional context in French, instead of a corpus of academic texts. Moreover, a bias is could be brought when comparing models trained on different datasets.

Thus, the objective here is to compare five different methods (Word2Vec SG and CBOW, GloVe, FasText SG and CBOW) to asses which of those model output the most accurate text representation.



They will be ranked regarding their ability to keep the semantic relationships between words found in the training corpus. We thus extended the related work by 1/ comparing the most recent and used embedding methods on their ability to preserve the semantic similarities between words, 2/ removing the bias brought by the utilization of different corpus to train the compared embedding methods and 3/ using these embedding algorithms on a challenging corpus instead of academic texts.

This representation will then be used as the input of deep learning models constructed to improve the annotating phase, actually performed by the ECMT in the SHDW. This NER phase will be the first step toward a multilingual and multi-terminologies concept extractor. Moreover, the constructed models will be the first available for the community working on medical documents in French through a public interface.

## Methods

### Corpus

The corpus used in this study is composed of a fraction of health documents stored in the SHDW of the RUH, France. All these documents are in French. They are also quite heterogeneous regarding their type: discharge summaries, surgery or procedure reports, drug prescriptions and letters from a general practitioner. All these documents are written by medical staff in the RUH and thus contain many typography mistakes, misspells or abbreviations. These unstructured text files were also cleaned by removing the common header (containing RUH address, phone numbers, etc).

### Documents de-identification

These documents were then de-identified to protect each identity of every patient or doctor from the RUH. Every first and last name stored in the RUH main databases were replaced by non-informative tokens such as <doctor>, <firstname> or <lastname>. Moreover, other tokens have been used such as <email> or <date>. In case of a misspelling of a patient's name in a document or of a lack in the database, a filter, based on REGular EXpressions (REGEX), has been defined to catch emails, doctors or professor names (based on prefix "Dr" or "Prof." respectively and their variations), abbreviations such "Mr" or "Mrs", dates and phone numbers without prior knowledge. To improve this important phase, a last rule has also been defined. If no patient or doctor name is found in the document, this text is just ruled out to prevent to release sensitive information in the embedding models.

### Pre-processing

First comes the question about the shape of the input data. Should it be composed of chunks of sentences (data is composed of a list of tokenized sentences) or sub-split by document (a list of tokenized documents)? The answer to this question depends on what the model will be used for. In our case, the context of each document is important (but not the context of each sentence, which is a good representation for documents dealing with many subjects). Therefore the input data will be based on document sub-splitting.

Then, the data has been lowered (no additional information would have been brought on words semantics similarity conservation by differences between upper and lower case for this study), the punctuation was removed and the numerical values were replaced by a meta-token <number>. We chose to not remove stopwords, due to their negligible impact on the context. Indeed, their multiple apparitions in many different contexts will just create a cluster of stopwords in the middle of the VSM.

## Training

The models have been implemented thank to the Gensim python library [24]. They have been trained on a server powered by four XEON E7-8890 v3 and 1To of RAM located on the RUH. We based the tuning of models' hyper-parameters on the literature [25] and on our own experience. The goal here being to compare word embedding implementation, we chose to keep equivalent parameters for each model. Chosen values are listed on the table 1.

Table 1. Hyperparameters used to train the five word embedding models.

epochs	Word2Vec / FastText	25
	GloVe	100
Minimum token count		20
Context window size		7
Learning rate		$2.5 \times 10^{-2}$
Embedding size		80
Alpha rate		0.05
Negative sampling	Word2Vec / FastText	12
subsampling	GloVe	$1e^{-6}$

## Evaluation

The goal behind these comparisons is to find the model that can represent a non-academic text into a mathematical form, which keep the contextual information about the words despite the bias brought by the poor quality of used language. To do so, different metrics have been defined, centered on word similarity tasks. The positive relationships are evaluated with the cosine similarity task and the negative ones with the odd-one task. Analogy-based operations and human evaluation allow to assess if a given model can keep the deep meaning of a token (antonyms, synonyms, hyponyms, hyperonyms, etc).

### Cosine similarity

Similarities between embedded pairs of concepts were evaluated by computing cosine similarity. It has also been used to assess whether two concepts are related or not. Cosine similarity (cos) between word vectors  $W_1$  and  $W_2$  indicates orthogonal vectors when close to 0 and highly similar vectors when close to 1. It is defined as:

$$\cos(W_1, W_2) = \frac{W_1 \cdot w_2}{\|W_1\| \cdot \|W_2\|}$$

It is possible to define a validation set, composed of couples of terms who should be used in a similar context in our documents (like "flu" and "virus"). Then, the first token from each couple is sent to each model and the top-ten closest vectors regarding the cosine similarity are extracted. The second word has to be retrieved in these 10 closest vectors to be considered as successful. Then, the total percentage  $p$  of success has been calculated regarding the total number of words pairs, with  $N$  being the number of times were the second term has been found in the top-ten closest vectors of the first one.

$$p = \frac{N \times 100}{N_{pairs}}$$

To construct the dataset, two well-known validation sets UMNSRS-Similarity and UMNSRS-Relatedness were used, containing 566 and 588 manually rated pairs of concepts respectively, known to be often found together [26]. However, our corpus being in French, the translated and aligned version of the MeSH terminology stored in HeTOP was used to translate these two sets [27]. The result provides a number of 308 pairs for the UMNSRS-Similarity and 317 for the UMNSRS-Relatedness, the remaining concepts weren't directly found in the MeSH.

### **Odd One Out similarity**

The odd one out similarity task tries to measure the model's ability to keep track of the words' negative semantic similarities by giving three different words to the model. Two of them are known as linked, not the third one. Then, the model has to output the word vector which does not clusterize with the two others (*e.g* output “*car*” when the input is “*car, basketball, tennis*”) [28]. To create such a validation corpus, every MeSH term appearing more than 1,000 times in the corpus has been extracted. The result was a list of 516 MeSH terms, which have been manually clusterized into 53 pairs of linked MeSH concepts according to two different Medical Doctors (MD). Then, 53 words appearing more than 1,000 times in the corpus have been randomly selected to be used as odd terms, one for each pair of MeSH term. The matrix of cosine distance between the three tokens was calculated for each item of the odd-one list and for each model. The goal for the model is to output a cosine distances between each of the two linked terms and the odd one closer to 0 compared to the one between those two linked terms, which should be closer to 1 (indicating more similar vectors). The percentage of success is then calculated following the same formula than for the cosine distance task.

### **Human evaluation**

A formal evaluation of the five methods was performed by a public health resident (CM) and a medical doctor (SJD). A list of 112 terms has been extracted from the Medical Sub Heading (MeSH) terminology. At least three concepts have been extracted from each branch of the MeSH terminology (regardless of branch Publication Characteristics, V). All of these 112 terms have been sent to each model, and the top-five closest vectors regarding the cosine distance have been extracted from every model. Overlapping top-close vectors between models were grouped, avoiding to evaluate several times the same answer, and the total list was randomized to avoid the annotator's tiredness. CM and SJD then blindly assessed the relevance of each vector compared to the sent token. These citations were assessed for relevance according to a three-modality scale used in other standard Information Retrieval test sets: bad (0), partial (1) or full relevance (2). Notes were also given blindly, annotators being not aware of which model output which top-close vector and the same set was revised twice (double-revision).

### **Analogy-based operations**

Mikolov's paper presenting Word2Vec showed that mathematical operation on vectors such as additions or subtractions are possible like the famous “(*king* – *man*) + *woman* ~ *queen*”. This kind of task helps to check the semantic analogy between terms. With the Mikolov's operation, it is possible to affirm that “*king*” and “*man*” share the same relationship properties than “*queen*” and “*woman*”. To check the conservation of these properties by each model, several mathematical operations covering a wide range of possible subjects found in the EHR (hospital departments, human tissues, biology, drugs) were defined following Mikolov's style (“(*Term 1* – *Term 2*) + *Term 3* ~ *Term 4*”). Then, the operation was performed using vectors “*Term 1*”, “*Term 2*” and “*Term 3*” extracted from each model. The resulting vector was compared to the “*Term 4*” vector, the operation being considered as correct if this “*Term 4*” vector is found to be the closest one regarding the cosine distance with the operation resulting one, indicating a semantic similarity between “*Term 3*” and “*Term 4*” similar to the one between “*Term 1*” and “*Term 2*”.

## Word clusters

In the VSM, words are grouped by semantic similarity, but the context does influence a lot this arrangement. Every model's vectors dimensions have been reduced and projected on two dimension using t-SNE algorithm. Then, logical words clusters have been manually searched in the projection. This step will not be part of the global final score, but allow to rapidly assess the quality of a word representation.

## Going further: model improvement

To check if a model pre-training affects or not the result, a new version of the best model regarding the tasks explained above will be trained two times. First, French articles abstracts from the LiSSa corpus (1,250,000 in total) will be used for model pre-training. Then, this resulting embedding will be trained a second time on the documents from the RUH without changing any parameters. All automatic tests will be performed against for this model a second time to assess if the added academic data improve the model's quality regarding our evaluation.

## Results

### Corpus

In total, 641,279 documents from the RUH have been de-identified and pre-processed. Regarding the vocabulary, texts have been split into 180,362,939 words in total, representing 355,597 unique tokens. However, this number can be pondered with 170,433 words appearing only one time in the entire corpus (mainly misspells, but also geographic locations or biological entities like genes, proteins, etc). In total, 50,066 distinct words are found more than 20 times in the corpus, thus present in the models (minimum count parameter set to 20). On average, each document contains 281.26 words (*standard deviation* (sd) = 207.42). The ten most common words are listed in table 2.

**Table 2.** The ten most common words of our corpus. Note that Rouen is the city where the training data comes from.

French	English	Occurrences
de	of	9,501,137
docteur	doctor	4,822,797
le	the	3,975,735
téléphone	phone	3,147,286
d'	's	3,036,198
Rouen	Rouen	2,763,918
à	at	2,271,317
l'	the	2,129,090
et	and	2,091,502
dans	in	2,001,135

These documents were decomposed using the Term-Frequency Inverse-Document-Frequency (TF-IDF) algorithm, which results in a frequency matrix. Each row, representing an article, has been used to clusterize those documents with a kMeans algorithm (number of classes  $\mathbf{K} = 5$ ). To visualize their distribution on two dimensions, t-SNE algorithm has been used (figure 1) [29].

**Figure 1.** Two-dimensional t-SNE projection of 10,000 documents randomly selected among main classes in the HDW. The five different colors correspond to the five types of documents selected (discharge summaries (green), surgery (blue) or procedure (purple) reports, drug prescriptions

(yellow), letters from a general practitioner (red)).



Those main classes are well separated, thus the vocabulary itself contained in the documents from the HDW is sufficient to clusterize each type of text. However, discharge summaries, surgery or procedure reports are a bit more mixed because of the words used in these kinds of context (short sentences, acronyms and abbreviations, highly technical vocabulary, etc). Regarding drug prescriptions and letters to a colleague or from a general practitioner, they present more specific vocabulary (drugs and chemicals, and current/formal language respectively), involving more defined clusters for these two groups.

## Training

Regarding the training time, models are very different. GloVe is the fastest algorithm to train with 18 minutes (min) to process the entire corpus. The second position is occupied by Word2Vec with 34 min and 3h02 (CBOW and SG architectures respectively). Finally, FastText is the slowest algorithm with a training time of 25h58 with SG and 26h17 regarding CBOW (Table 3).

Table 3. Algorithms training time (minutes). GloVe is the fastest algorithm to train.

Algorithm	Training time (min)
FastText SG	1678.1
FastText CBOW	1577.0
Word2Vec SG	182.0
Word2Vec CBOW	33.4
<b>GloVe</b>	<b>17.5</b>

GloVe performs much better in terms of computational time due to the way it handles the vocabulary. It is stored as a huge co-occurrence matrix and thanks to its count-based method, which is not computationally heavy, it can be highly parallelized. It was expected that FastText would take a lot of time to train, due to the high number of words sub-vectors it creates. However, for Word2Vec, the difference between the two available sub-architectures is highly visible (33 min to 3h02). This difference could come from the hierarchical soft-max and one-hot vector used by the CBOW architecture, which reduces the usage of the CPU. With SG, the minibatch parsing of all the *context || target* pairs highly increases the time to go through all possibilities.

## Evaluation

### *Cosine similarity*

The total number of UMNSRS pairs successively retrieved by each model has been extracted (308+317 pairs in total with UMNSRS-Rel and UMNSRS-Sim). The percentages of validated pairs from the UMNSRS datasets are presented in the table 4. FastText SG performed this task with the highest score (3.89% and 5.04% for UMNSRS-Sim and UMNSRS-Rel respectively). The very low scores indicate that this kind of published dataset is useful to validate models trained on more academic texts.

**Table 4.** Percentage of pairs validated by the five trained models on two UMNSRS evaluation sets.

Algorithm	UMNSRS-Sim	UMNSRS-Rel
<b>FastText SG</b>	<b>3.89</b>	<b>5.04</b>
FastText CBOW	3.89	3.79
Word2Vec CBOW	3.57	4.10
Word2Vec SG	2.92	4.10
GloVe	1.29	0.94

### *Odd one similarity*

Regarding the odd one similarity task, models are quite different (table 5). Word2Vec is the best so far with 65.4% and 63.5% of odd one terms correctly isolated with SG and CBOW architectures respectively. Both FastText architectures achieve a score between 44.4% (SG) and 40.7% (CBOW). GloVe only found the correct odd terms in 18.5% of the tested tasks.

**Table 5.** Percentage of odd one tasks performed by each of the five trained models.

Algorithm	OddOne
<b>Word2Vec SG</b>	<b>65.4</b>
Word2Vec CBOW	63.5
FastText SG	44.4
FastText CBOW	40.7
GloVe	18.5

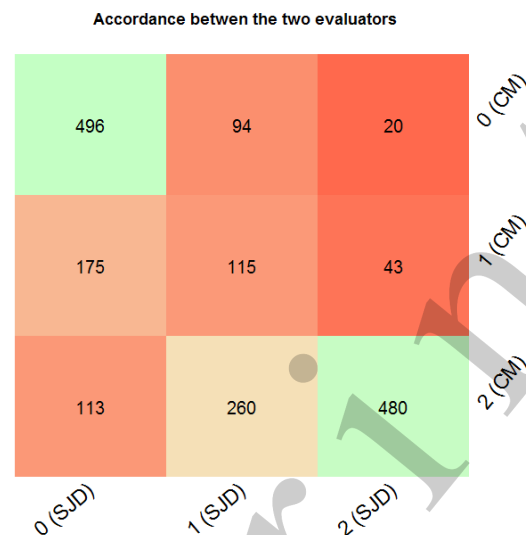
Regarding the sub-architectures presented by both Word2Vec and FastText, the SG always performed better than the CBOW, possibly due to the negative sampling. Indeed, the studied corpus is quite heterogeneous and words can be listed as items (drugs, *e.g*) instead of being used in correct sentences. Thus sometimes, the complete update of vectors' dimensions generates non-senses in the models (items from lists are seen as adjacent by the models, thus used in same sentences, leading to non-senses).

### *Human validation*

The evaluation focused on 1,796 terms (5 vectors \* 112 MeSH concepts \* 5 models and 1,004 were returned multiple times by different models) rated from 0 to 2 by two evaluators. First, the agreement between CM and SJD was assessed with a weighted kappa test [30]. A kappa  $k = 0.6133$  was obtained. According to the literature, the agreement between the two evaluators can be considered as substantial [31]. This agreement can be retrieved in figure 2. The accord is stronger for the extreme

scores (0 and 2) while the agreement about the middle score of 1 is least pronounced.

**Figure 2.** Global representation of the notation agreement between the two evaluators (CM and SJD). Notes attributed to a model output are going from 0 (bad matching) to 2 (good matching). Colors are ranging from light green (high agreement) to red (low agreement).



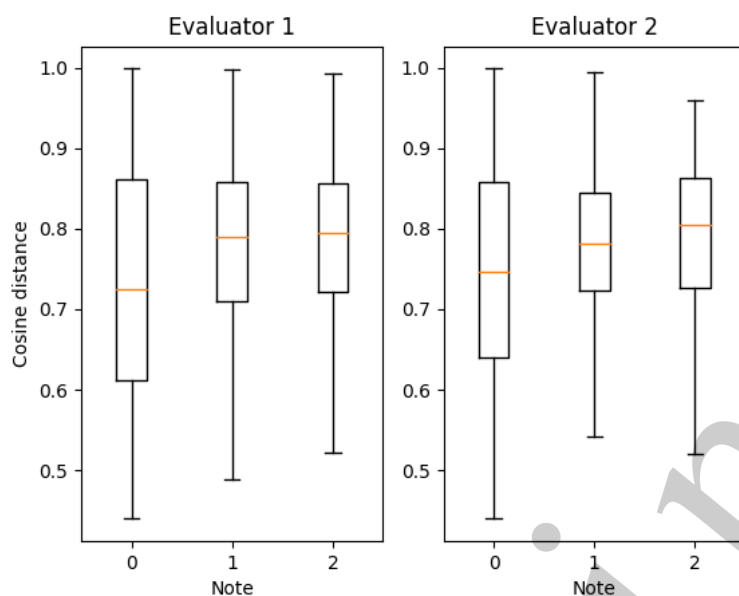
Moreover, to assess if human evaluators remained coherent regarding the cosine distance computed by each model, the average note given by the two evaluators was compared with the average of the cosine distance computed for each model (Table 6). Word2Vec with the SG architecture performed the highest score, regardless of the evaluator (1.469 and 1.200). Interestingly, GloVe computes the closest to 1 cosine distance in averages (0.884 on the top-five terms to each of the 112 given concepts, indicating the highest similarity), while both evaluators gave it the lowest grade.

**Table 6.** Comparison between cosine distance computed by each model and the human evaluation performed (notes ranging from 0 to 2). Notes and distances are in averages on the top-5 closest vectors for 112 queries on every model by each of the two evaluators (evaluator 1, SJD; evaluator 2, CM).

Model	Cosine	Evaluator 1	Evaluator 2
Word2Vec SG	0.776	<b>1.469</b>	<b>1.200</b>
Word2Vec CBOW	0.731	1.355	1.148
FastText SG	0.728	1.200	1.111
FastText CBOW	0.748	1.214	1.048
GloVe	<b>0.884</b>	0.925	0.480

To go further, the cosine distances between the 112 sent concepts and the 1,796 returned were plotted for each of the three modalities rated by the evaluators (Figure 3). In fact, when humans are judging the quality of a returned vector as poor (note 0), the cosine distance between this vector and the queried one is also lower and vice-versa.

**Figure 3.** Comparison of the cosine distance calculated regarding the note given by two human evaluators. In both cases, the lower the note is, the lower the average distance is (evaluator 1, SJD; evaluator 2, CM).



### Analogy-based

#### operations

A list of six mathematical operations has been defined with the help of a MD and a university pharmacist (listed in table 7). Each operation consists in verifying if  $(\overrightarrow{Term\ 1} - \overrightarrow{Term\ 2}) + \overrightarrow{Term\ 3} \approx \overrightarrow{Term\ 4}$ , allowing to check if the similarity between “Term 1” and “Term 2” is the same than the one between “Term 3” and “Term 4”, which should be. These operations have been defined to cover a wide range of subjects (RUH departments, drugs, biology, etc).

**Table 7.** Logical operations on words having to be retrieved with the different trained models. For each, first line: French, second one: English. Relation 1 describes anatomical/medical relationship, relation 2 is for cancer/location, relation 3 for globule/functions, relation 4 treats tissues/drugs relationships, relation 5 the anatomy/location and finally relation 6 describes the drugs/effects links.

1	(cardiologie - coeur) + poumon ~ pneumologie
	(cardiology - heart) + lung ~ pneumology
2	(mélanome - peau) + glande ~ adénome
	(melanoma - skin) + gland ~ adenoma
3	(globule - sang) + immunitaire ~ immunoglobuline
	(corpuscle - blood) + immune ~ immunoglobulin
4	(rosémide - rein) + coeur ~ fosinopril
	(furosemide - kidney) + heart ~ fosinopril
5	(membre - inférieur) + supérieur ~ bras
	(limb - lower) + upper ~ arm
6	(morphine - opioïde) + antalgique ~ perfalgan
	(morphine - opioid) + analgic ~ perfalgan



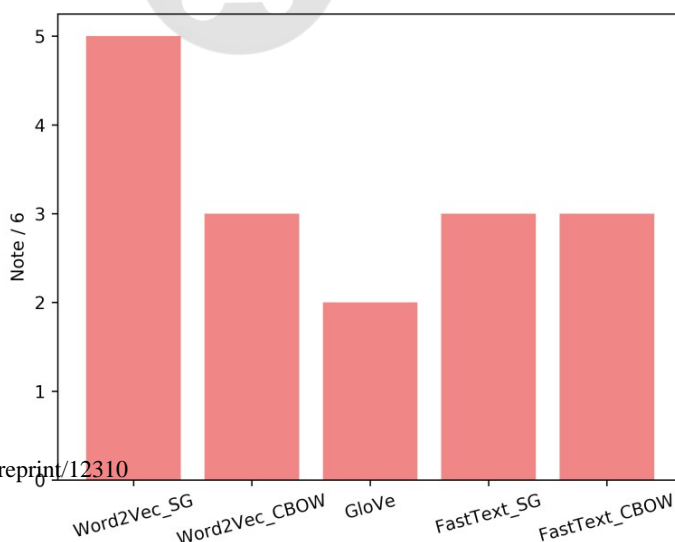
Each operation  $(\overrightarrow{Term\ 1} - \overrightarrow{Term\ 2}) + \overrightarrow{Term\ 3}$  has been performed on vectors from each model and the nearest vector to the resulting one has been extracted. Score for each model is presented on figure 4. Word2Vec got the highest score on this task, especially for SG architecture (5/6), while GloVe got the lowest one (2/6).

Interestingly, no operation has been failed by the five models, indicating that none of them is simply not logical or just too hard to perform for word embedding models. Operation 2 has been missed by both Word2Vec and FastText SG while CBOW architectures success to perform it for both algorithms. In the corpus, tumors ("*mélanome*" ("*melanoma*") and "*adénome*" ("*adenoma*")) are cited far from their localization ("*peau*" ("*skin*") and "*glande*" ("*gland*") respectively). This distance may be too high for the context-window size (7 words).

GloVe only performed operations 1 and 5. Only Word2Vec SG succeeds on the 5th one. The low score for this task can come from the fact that GloVe treats only pairs of words in the co-occurrence matrix. Thus, relations in common between two tokens and a third one are not taken in account.

FastText algorithm just got the average score with SG and CBOW. They both failed to perform operations number 4 and 5 (also number 2 for SG and number 3 for CBOW). The sub-word decomposition performed by this algorithm is keeping track of the context, but is not as accurate as Word2Vec SG in this task. This imbalance is not compensated by the SG architecture, which performed better for Word2Vec, indicating that this sub-word decomposition has a really strong impact on the embedding.

**Figure 4.** Score for mathematical operations task on six point maximum for each of the five trained models. Word2Vec is the best so far with a score of 5/6.



## Word clusters

As a visual validation, t-SNE algorithm was applied on vectors extracted from every of the five models. To investigate how word vectors are arranged, clusters have been manually searched on the projection. Word2Vec is clustering words with a good quality regarding the context they can be used in. Both SG and CBOW architectures have logical word clusters, for example related to time (Figure 5).

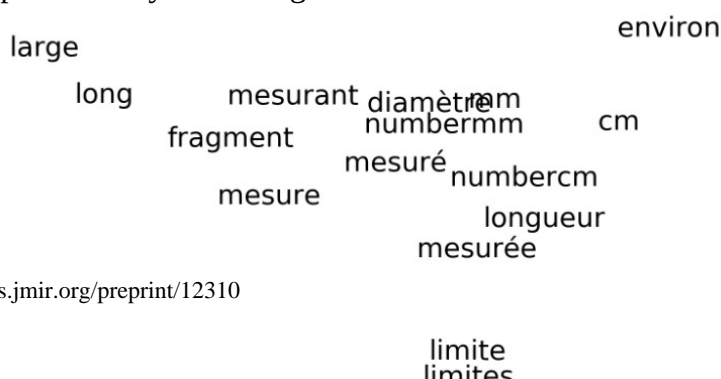
Figure 5. Small cluster of words found in both Word2Vec SG and CBOW (second one shown). *Année(s)* and *an(s)* mean *year(s)*, *semaine(s)* mean *week(s)* and *jour(s)* mean *day(s)*. The meta-token "number" used to replace numbers is visible in the expression "numberj".



Many other clusters are found by reducing the dimension of both Word2Vec SG and CBOW results; some are showed on supplementary Figure 1. These clusters of linked words are underlying the fact that the context on which words are used has a strong impact on the words vectorization for this algorithm. On the figure 5, it is easily visible that the word structure itself (word size, letters composing it, etc) does not influence at all the representation of words produced by Word2Vec. In fact, tokens seen in this insert are very different, regarding the size (ranging from two letters for "an" ("year") to eight for "semaines" ("weeks") or the composition of letters (no letters in common between the two neighbors "semaine" ("week") and "jour" ("day")).

By looking at the dimensional reduction of vectors produced by GloVe, it is visible how co-occurrence matrix used by this algorithm is affecting the placement of vectors in the VSM. In fact, words often used close to each other's (and not especially on the same context, like Word2Vec) are clusterizing well. In the group given as example on the figure 6, it is visible that sentence segments are almost found intact. Indeed, the large co-occurrence matrix capture well similarities found inside the sliding window, but two words having the same meaning but not found in the same context (*i.e* surrounded by different other tokens) will have more difficulties to clusterize with this algorithm.

Figure 6. Cluster of words related to the size found by reducing the number of dimensions of word vectors produced by GloVe algorithm.



Regarding FastText, it is interesting to notice that clusters of words used in a similar context are found but others variables does influence a lot the spatial arrangement of the vectors while projected on 2 dimensions: word size and composition. Indeed, as seen on the supplementary figure 2, a gradient starting from the edges of the word projection to the center is following the size of tokens. The shortest ones are found on the edges while the longest in the middle, indicating than the sub-word vectors created by FastText to decompose each word are strongly impacted by the morphological structure of embedded words.

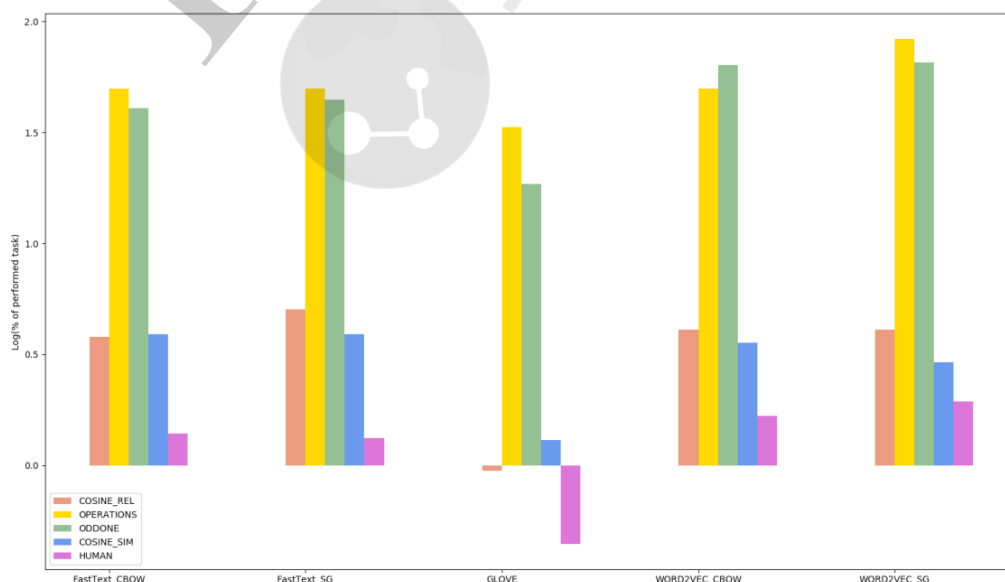
Regarding the global shape of the five projections on the supplementary figure 3, no meaningful distinction can be made between the five studied models at this scale. The diversity found at a local scale is not projected on the global one.

### Model improvement

So far, Word2Vec with the Skip-Gram architecture showed the best results in average (figure 7). Thus, a subset of 350,000 French abstracts has been extracted from the LiSSa database, hosted at the RUH, to pre-train this embedding model. It took nearly 20 minutes to the algorithm to pre-process these data with the same workflow than the one presented in method section and to train on it (parameters listed on table 1). Afterwards, another 48 minutes were needed to update word vectors thanks to the 607,135 health documents contained in the HDW from the RUH.

When this model trained on two different dataset is compared to the initial word2vec model (without any pre-training), scores are not changed regarding the cosine and the odd one tests (4.1% on the UMNSRS-Rel and 65.4% respectively). Interestingly, the grade coming from analogy-based operations decreases, lowered from 5/6 to 3/6. This can come from the fact that documents used for pre-training (scientific articles) are highly specialized in a domain, leading to already strongly associated vectors.

**Figure 7.** Pulled scores for each task regarding every of the five trained models. Log has been used to facilitate the visualization. Cosine score is duplicated regarding the UMSNRS used set.



## Discussion

### Principal Results

In this study, the three most famous word embeddings have been compared on a corpus of challenging documents (two architectures each for Word2Vec and FastText as well as GloVe) with five different evaluating tasks. The positive and negative semantic relationships have been assessed, as well as the word sense conservation by human and the analogy-based evaluation.

The training in our 600k of challenging documents showed that Word2Vec SG got the best score for three on the four rated tasks (FastText SG is the best regarding the cosine one). These results are coherent with those obtained by Th *et al.*, who compared Word2Vec and GloVe with the cosine similarity task [32]. More specifically, the CBOW architecture is training way faster while the SG is more accurate on semantic relationships. This model seems to be more influenced by the context in which each word is used than by the word composition itself. GloVe got the worst grade regarding to our evaluations, however it is the fastest to train so far. Moreover, GloVe was the only one not implemented in the Python library *Gensim*, which could have brought a bias in this study. This model is computing a cosine distance closer to 1 in average between queried word and close ones, while the human judgment shows the lowest grade. Regarding FastText, it is interesting to notice that the morphological similarities are kept in account in the vector space creation. In fact, word clusters are highly impacted by the word's composition in letters and by its size. However, the sub-vector decomposition of words allows this kind of model to be queried by words absent from the original training corpus, which is impossible with others. Therefore, this model could be used for orthographic correction or acronym disambiguation for example.

The medical corpus used as a training set for these embedding models is coming from a real work environment. First, finding a good evaluation for embeddings produced in such a context is a hard task. The performances shown by some models trained on scientific literature or on other well-written corpus should be biased regarding their utilization on a very specific work environment. Second, based on our results, an amount of 26.1% of unique tokens found in the health-related documents are not present in an academic corpus of scientific articles, indicating a weakness of the pre-trained embedding models. Documents produced in a professional context are highly different compared to this kind of well written texts. Finally, in this study, pre-training an embedding with an academic corpus then on the specific one does not improve the model's performances. It even lower the score associated to analogy-based operations, indicating strongly associated vectors in the VSM, which lead to a loose of the inherent plasticity of this kind of model to deeply understand the context of a word.

There are a few limitations in our study. First, other embedding models, newly released, could have been compared as well (BERT [33], ELMo [33], etc). Second, other clinical notes from different health establishments could have been joined to this study, to investigate how the source of the corpus could affect the resulting similarities found in the embedding space. The complete comparison could also have been trained on non-clinical data, which are highly sensitive and hard to obtain, to help reproducibility. Finally, the quality of those embedding has been checked regarding the semantic similarity conservation, but other metrics could affect this judgment, depending on the model's usage.

Regarding the cosine annotation, low scores could be explained by the number of occurrences of each term from the 625 words pairs in the corpus of texts. The UMNSRS-Rel dataset contains 257 unique terms for 317 word pairs, while the UMNSRS-Sim contains 243 terms for 308 pairs. First, 128 words in total (25.6%) have been found less than 20 times regarding all of the 641,279 documents, thus being absent in the model due to the "*min\_count*" parameter. These words are found in 452 word pairs in total (231/317 in the UMNSRS-Rel and 221/308 in the UMNSRS-Sim), representing 72.3% of the total number of word pairs searched in the models who cannot be found.

Most of these words absents from the models are drugs molecular names, while practitioners from the RUH often use the trade name to refers to a drug (eg. ESPERAL © instead of *disulfirame*). The natural medical language used in the RUH by the practitioners prevents some words to be found: use of an acronym ("*HTA*" instead of "*hypertension artérielle*", meaning "*hypertension*") or of a synonym ("*angor*" instead of "*angine de poitrine*", meaning "*angina pectoris*"). Another explanation could come from the fact that some associations defined in those UMNSRS datasets can be true in an academic context, but will be very rarely found in a professional context.

With a median number of occurrences of 230 in the entire corpus of health documents, 176 words (28.1%) have been found more than 1,000 times. While the biggest proportion of the low-frequency words was composed of drugs or molecules names, the high-frequency group of words (up to 134,371 times for the word "*douleur*", meaning "*pain*") is mainly composed of clinical symptoms or diseases. This validation corpus seems to be just not suitable to investigate the quality of embedding trained on such a corpus.

## Conclusions

In our case, Word2Vec with the SG architecture got the best grade regarding three out of the four rated tasks. This kind of embedding seems to preserve the semantic relationships existing between words and will soon be used as the embedding layer for a deep learning based semantic annotator. More specifically, this model will be deployed for semantic expansion of the labels from medical controlled vocabularies. To keep the multi-lingual properties of the actual annotator, a method of alignment between the produced embedding and other languages will also be developed. Other tested recent unsupervised embedding method exhibit certain a quality, but their ability to preserve the semantic similarities between words seems weaker or influenced by other variable than word context.

As soon as the paper is submitted, any end user will be able to query the word embedding models produced by each method on a dedicated web site as well as to download high quality dimension reduction images and test sets [35]. This embedding will be the first publicly available, allowing the NLP medical research on French language to go further.

## Acknowledgements

This work was partially granted by the Ph.D CIFRE number 2017/0625 from the French Ministry of Higher Education and Scientific Research and by the OmicX company (ED). The authors thank Catherine Letord, pharmacist, Jean-Philippe Leroy, M.D. for their help in creating the test datasets and Pr. Xavier Tannier for the critical read-through.

ED developed algorithms, made statistics and drafts the manuscript. RL, CM and GK helped to create the test datasets and to evaluate the models. BD and JG helped with servers' utilization. SC and SJD supervised the study.

## Conflicts of Interest

The authors declare no conflict of interest.

## Abbreviations

CBOW: continuous bag-of-words

DRGs: diagnosis related groups

EHR: electronic health records

HDW: health data warehouse

LSA: latent semantic analysis

NER: named entity recognition

NLP: natural language processing

RUH: Rouen university hospital

SG: skip-gram

VSM: vector space model

## References

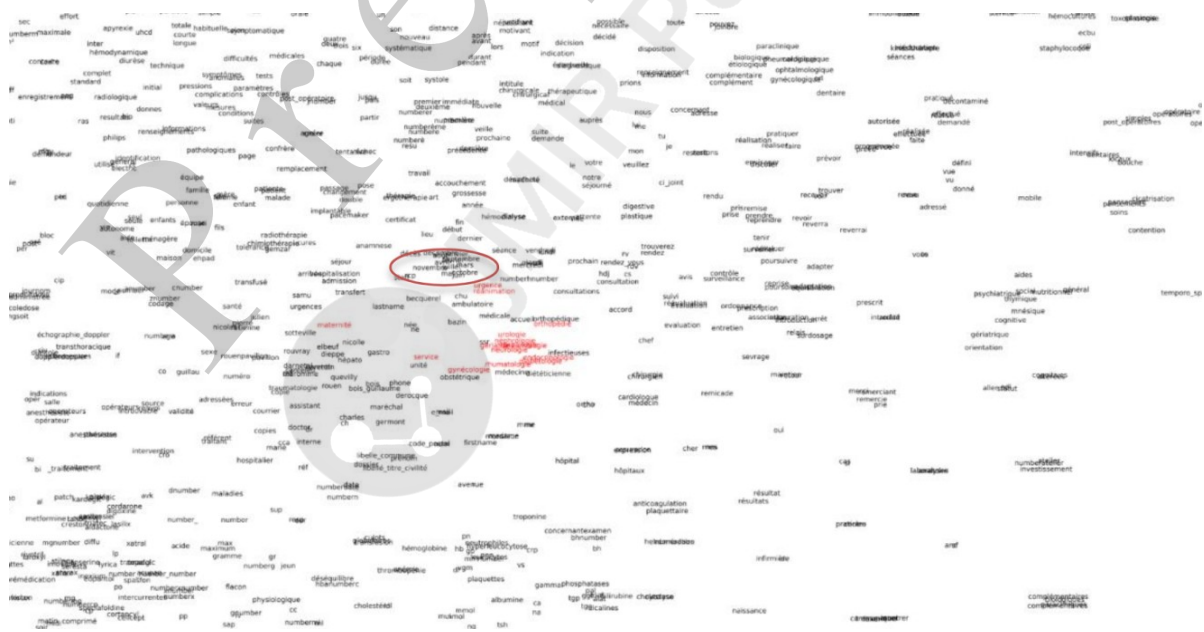
1. Grosjean J, Merabti T, Dahamna B, Kergourlay I, Thirion B, Soualmia LF, Darmoni SJ. Health multi-terminology portal: a semantic added-value for patient safety. *Stud Health Technol Inform* 2011;166:129–138. PMID:21685618
2. Tvardik N, Kergourlay I, Bittar A, Segond F, Darmoni S, Metzger M-H. Accuracy of Using Natural Language Processing Methods for Identifying Healthcare-associated Infections. *Int J Med Inf* 2018;117:96–102. PMID:30032970
3. Lelong R, Soualmia L, Dahamna B, Griffon N, Darmoni SJ. Querying EHRs with a Semantic and Entity-Oriented Query Language. *Stud Health Technol Inform* 2017;235:121–125. PMID:28423767
4. Firth JR. *A Synopsis of Linguistic Theory*. Basil Blackwell Oxf 1957;59:168–205.
5. Salton G. The SMART retrieval system-experiments in automatic document processing. *IEEE Trans Prof Commun* 1971;15:17–17. [doi: 10.1109/TPC.1972.6591971]
6. Singhal A. *Modern Information Retrieval: A Brief Overview*. *EEE Comput Soc Tech Comm Data*

- Eng 2018;24:35–43. [doi: 10.1.1.117.7676]
7. Baroni M, Dinu G, Kruszewski G. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. Proc 52nd Annu Meet Assoc Comput Linguist 2014;1:238–247. [doi: 10.3115/v1/P14-1023]
  8. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed Representations of Words and Phrases and their Compositionality. NIPS13 Proc 26th Int Conf Neural Inf Process Syst 2013;2:3111–3119. [doi: arXiv:1310.4546]
  9. Bengio Y, Ducharme R, Vincent P, Jauvin C. A Neural Probabilistic Language Model. J Mach Learn Res 2003;3:1137–1155. [doi: 10.1109/72.846725]
  10. Rong X. word2vec Parameter Learning Explained. arXiv 2014; [doi: arXiv:1411.2738]
  11. Guthrie D, Allison B, Liu W, Guthrie L, Wilks Y. A Closer Look at Skip-gram Modelling. Proc Fifth Int Conf Lang Resour Eval 2006;
  12. Goldberg Y, Levy O. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv 2014; [doi: arXiv:1402.3722]
  13. Pennington J, Socher R, Manning CD. GloVe: Global Vectors for Word Representation. Conf Proc 2014 Conf Empir Methods Nat Lang Process 2014;14:1532–1543. [doi: 10.3115/v1/D14-1162]
  14. Kolda TG, O'Leary DP. A semidiscrete matrix decomposition for latent semantic indexing information retrieval. ACM Trans Inf Syst 1998;16:322–346. [doi: 10.1145/291128.291131]
  15. Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching Word Vectors with Subword Information. Trans Assoc Comput Linguist 2017;5:135–146. [doi: arXiv:1607.04606]
  16. Joulin A, Grave E, Bojanowski P, Mikolov T. Bag of Tricks for Efficient Text Classification. Proc 15th Conf Eur Chapter Assoc Comput Linguist 2017;2:427–431. [doi: 10.18653/v1/E17-2068]
  17. Scheepers T, Gavves E, Kanoulas E. Analyzing the compositional properties of word embeddings. Univ Amst 2017;
  18. Papineni K, Roukos S, Ward T, Zhu W-J. BLEU: a method for automatic evaluation of machine translation. Proc 40th Annu Meet Assoc Comput Linguist 2001;40:311–318. [doi: 10.3115/1073083.1073135]
  19. Bairong Z, Wenbo W, Zhiyu L, Chonghui Z, Shinozaki T. Comparative Analysis of Word Embedding Methods for DSTC6 End-to-End Conversation Modeling Track. Tokyo Inst Technol 2016; [doi: arXiv:1706.07440]
  20. Beam AL, Kompa B, Fried I, Palmer NP, Shi X, Cai T, Kohane IS. Clinical Concept Embeddings Learned from Massive Sources of Multimodal Medical Data. arXiv 2018; [doi: arXiv:1804.01486]
  21. Huang J, Xu K, Vydiswaran VGV. Analyzing Multiple Medical Corpora Using Word Embedding. 2016 IEEE Int Conf Healthc Inform ICHI 2016 Oct;2016:527–533. [doi: 10.1109/ICHI.2016.94]

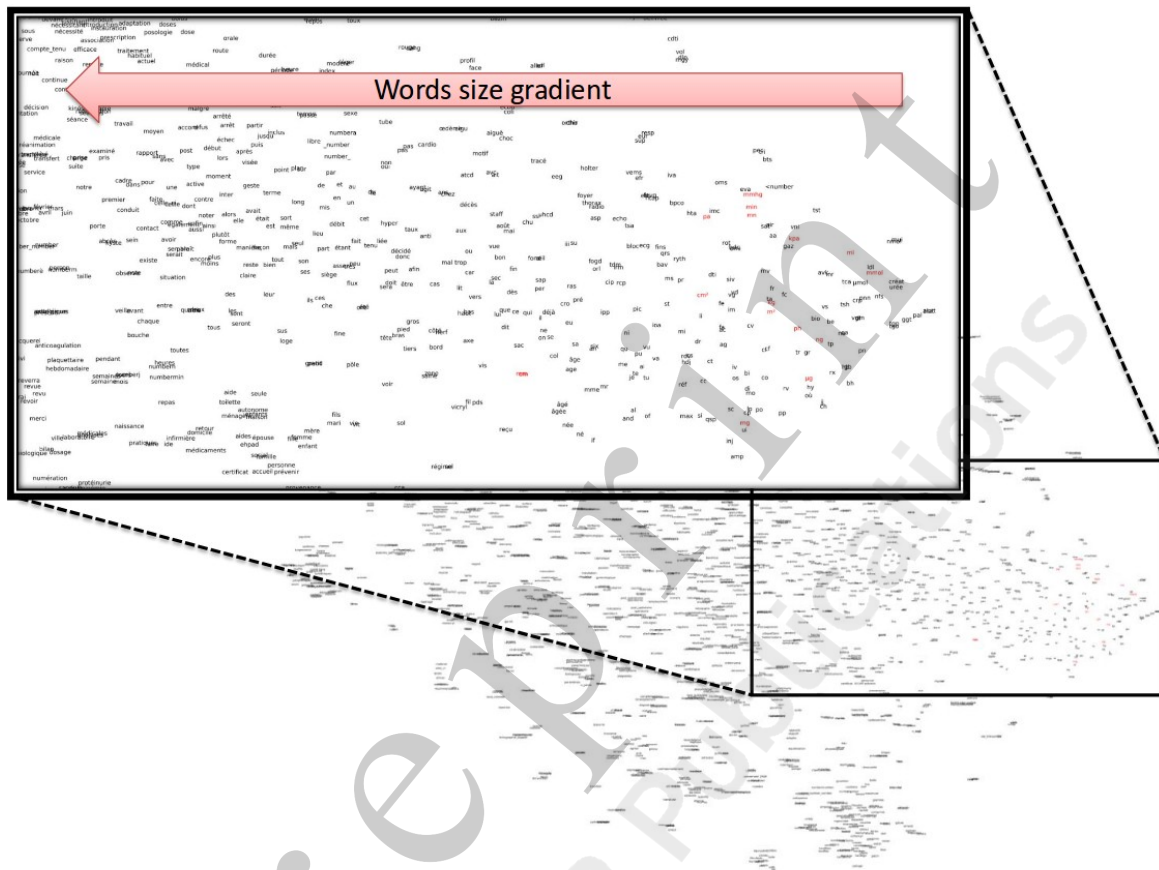
22. Wang Y, Liu S, Afzal N, Rastegar-Mojarad M, Wang L, Shen F, Liu H. A Comparison of Word Embeddings for the Biomedical Natural Language Processing. *J Biomed Inform* 2018;85. PMID:30217670
23. Griffon N, Schuers M, Darmoni SJ. LiSSa: An alternative in French to browse health scientific literature? *Presse Med* 2016;45:955–956. PMID:27871426
24. Rehurek R, Sojka P. Software Framework for Topic Modelling with Large Corpora. *Proc Lrec 2010 Workshop New Chall Nlp Framew 2010*. p. 45–50.
25. Chiu B, Crichton G, Korhonen A, Pyysalo S. How to train good word embeddings for biomedical NLP. *Proc 15th Workshop Biomed Nat Lang Process 2016*;15:166–174. [doi: 10.18653/v1/W16-2922]
26. Pakhomov S, McInnes B, Adam T, Liu Y, Pedersen T, Melton GB. Semantic similarity and relatedness between clinical terms: an experimental study. *AMIA Annu Symp Proc 2010*;2010:572–576. PMID:21347043
27. CISMef. HeTOP [Internet]. [cited 2018 Sep 25]. Available from: <https://www.hetop.eu/hetop/>
28. Sinapov J, Stoytchev A. The odd one out task: Toward an intelligence test for robots. *IEEE 9th Int Conf Dev Learn 2010*;9. [doi: 10.1109/DEVLRN.2010.5578855]
29. Maaten L van der, Hinton G. Visualizing Data using t-SNE. *J Mach Learn Res* 2008;9(Nov):2579–2605.
30. Cohen J. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychol Bull* 1968;70:213–220. PMID:19673146
31. McHugh ML. Interrater reliability: the kappa statistic. *Biochem Medica* 2012;22:276–282. PMID:23092060
32. Th M, Sahu S, Anand A. Evaluating distributed word representations for capturing semantics of biomedical concepts. *Proc BioNLP 15 Association for Computational Linguistics*; 2015. p. 158–163. [doi: 10.18653/v1/W15-3820]
33. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* 2018; [doi: arXiv:1810.04805]
34. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L. Deep contextualized word representations. *arXiv* 2018; [doi: arXiv:1802.05365]
35. Dynamant E. URL: <https://cispro.chu-rouen.fr/winter/> [Internet]. [cited 2018 Sep 25]. Available from: <http://www.webcitation.org/72h2BtgqL>



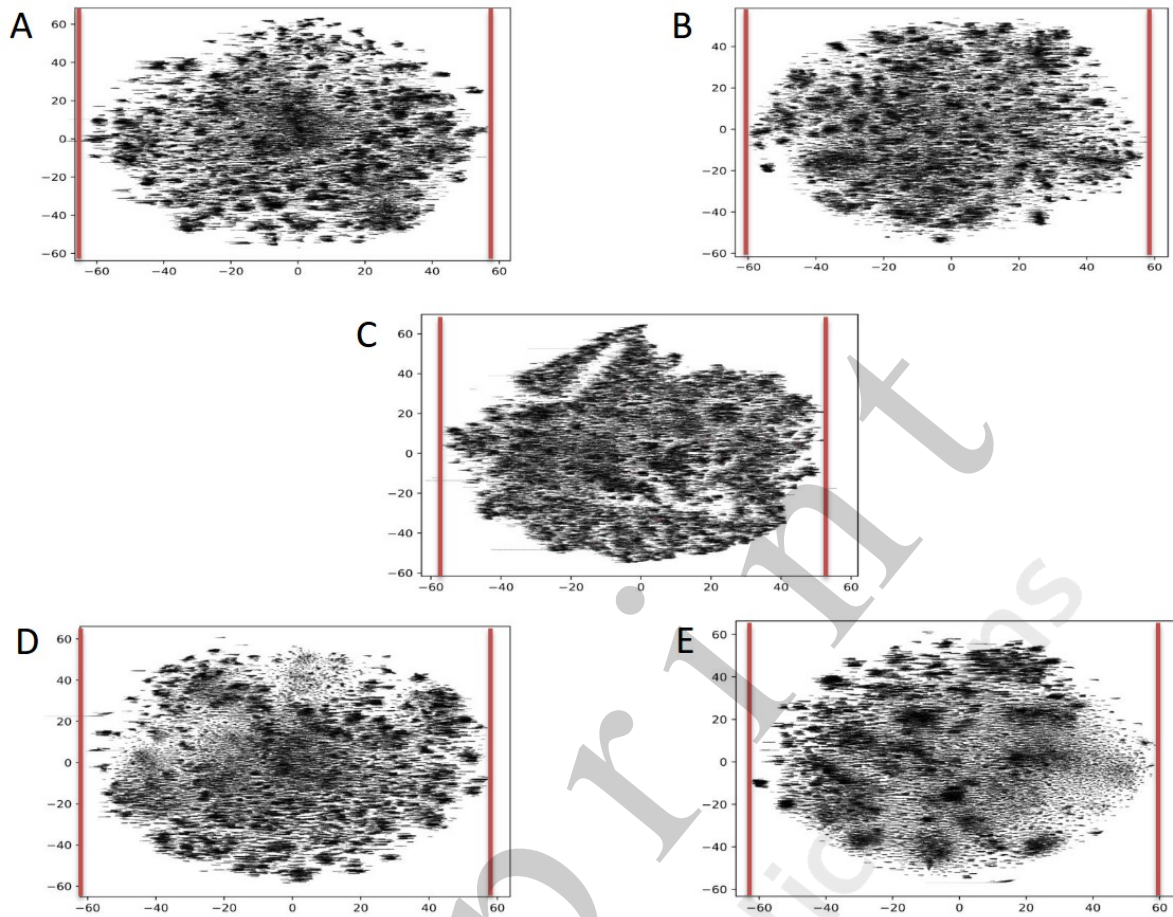
Supplementary figure 1. Other word clusters found in the Word2Vec model (CBOW architecture). Red words represent departments from the RUH (cardiology, gynecology, pneumology, etc.) while the red circle indicate months of year. These two groups are near because of the appointment letters or the summary of patients' medical background found in the corpus. Only words appearing more than 5,000 times in the entire corpus have been plotted.



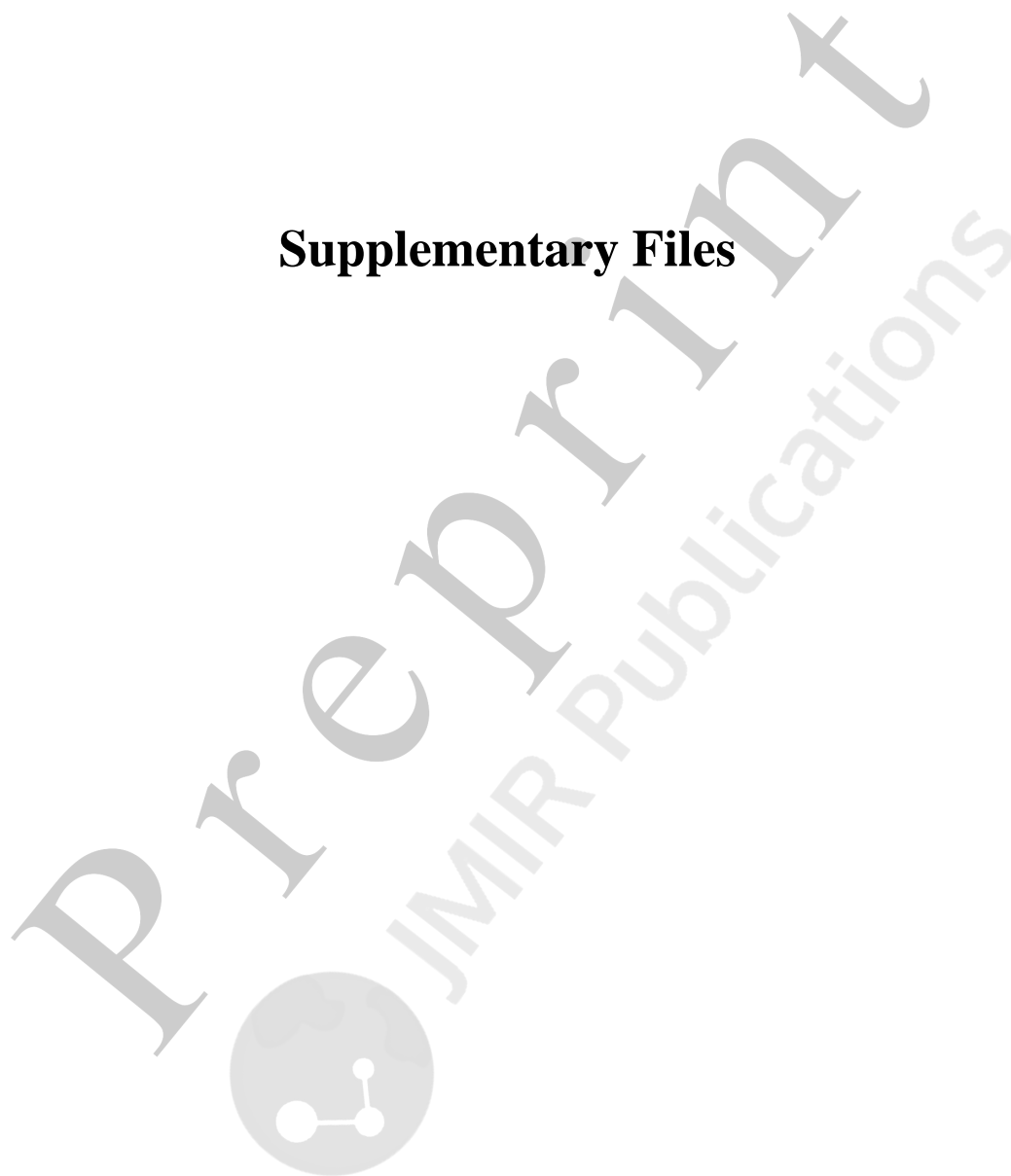
Supplementary figure 2. Words size gradient visible while projecting FastText model in two dimensions. In the background is the entire model, in the front the middle-right squared piece zoomed. Red words correspond to units for International Systems. They are grouped with two or three-letters words, while words visible on the left are longer. Only tokens appearing more than 5,000 times in the entire corpus have been plotted.



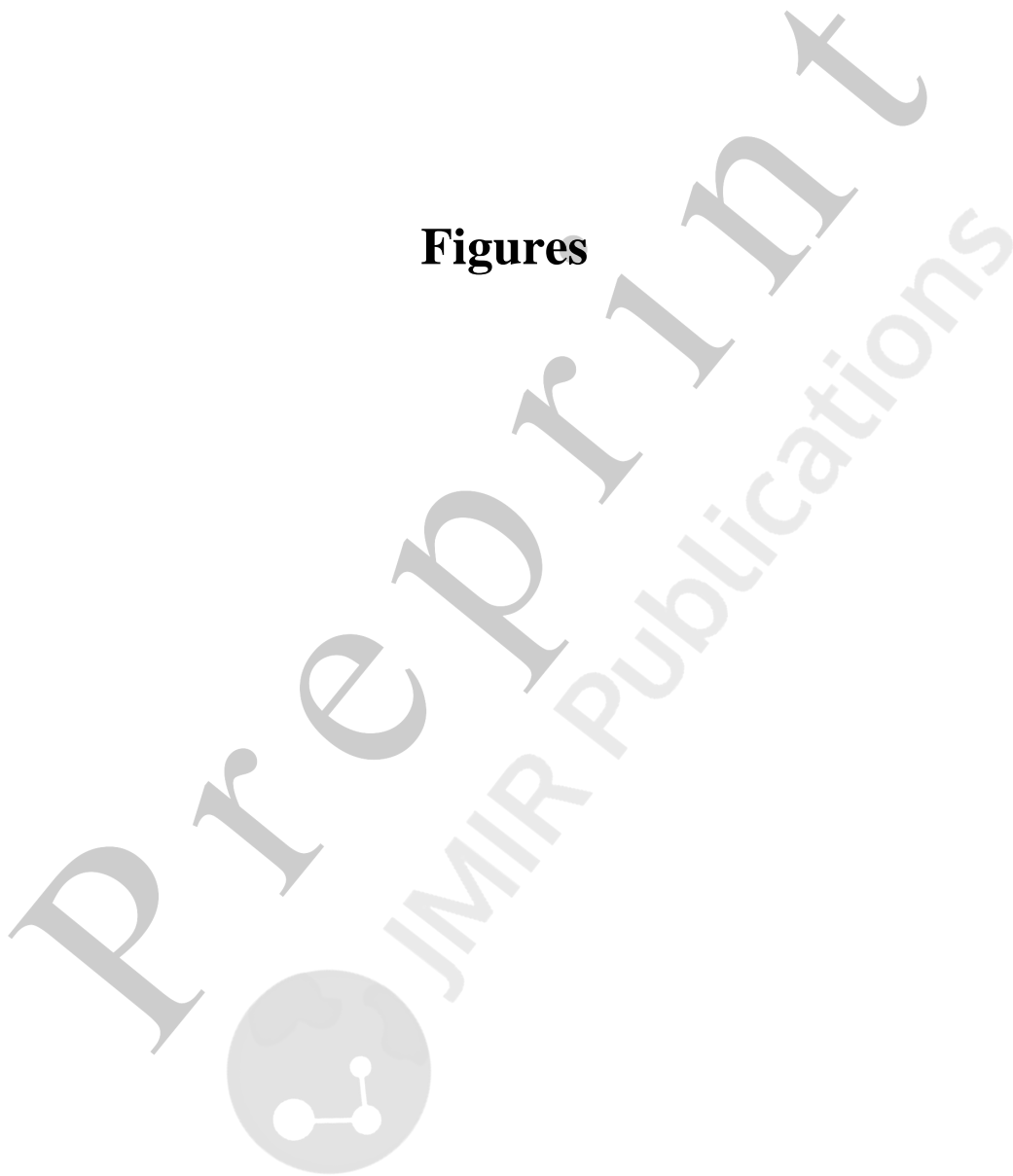
Supplementary figure 3. Global shape of the cloud generated by the dimension reduction by t-SNE of the five VSM created by the five trained word embedding models. Clouds design is highly similar; however, Word2Vec CBOW (figure B) seems to be more compact regarding the y axis compared to the other four. A: Word2Vec SG; B: Word2Vec CBOW; C: GloVe; D: FastText SG; E: FastText CBOW.



## Supplementary Files



## Figures



**Figure 4.** Score for mathematical operations task on six point maximum for each of the five trained models. Word2Vec is the best so far with a score of 5/6.

