MEDINFO 2019: Health and Wellbeing e-Networks for All
L. Ohno-Machado and B. Séroussi (Eds.)

doi:10.3233/SHTI190195

Word Embedding for French Natural Language in Healthcare: A Comparative Study

Emeric Dynomant[a,b,c], Romain Lelong[b,c], Badisse Dahamna[b,d], Clément Massonnaud[b], Gaëtan Kerdelhué[b,d], Julien Grosjean[b,d], Stéphane Canu[c], Stéfan Darmoni[b,d]

[a] *OmicX, 72 Rue de la République, 76140, Le Petit Quevilly, Normandie, France,*
[b] *Department of Biomedical Informatics, Cour Leschevin, CHU de Rouen, 1 Rue de Germont, 76031 Rouen, Normandie, France*
[c] *LITIS, Université de Rouen Normandie, Avenue de l'Université, 76800, Saint-Étienne-du-Rouvray, Normandie, France*
[d] *LIMICS, Campus des Cordeliers, INSERM U1142, 15 Rue de l'École de Médecine, 75006, Paris, France*

Abstract

Structuring raw medical documents with ontology mapping is now the next step for medical intelligence. Deep learning models take as input mathematically embedded information, such as encoded texts. To do so, word embedding methods can represent every word from a text as a fixed-length vector. A formal evaluation of three word embedding methods has been performed on raw medical documents. The data corresponds to more than 12M diverse documents produced in the Rouen hospital (drug prescriptions, discharge and surgery summaries, inter-services letters, etc.). Automatic and manual validation demonstrates that Word2Vec based on the skip-gram architecture had the best rate on three out of four accuracy tests. This model will now be used as the first layer of an AI-based semantic annotator.

Keywords:

Natural language processing, word processing, data mining.

Introduction

Context

The use of clinically derived data from electronic health records and other clinical information systems can greatly facilitate clinical research as well as optimizing diagnosis related groups, operational and quality initiatives. The main approach for making this data available is to incorporate the data from different sources into a joint health data warehouse that contains different kinds of natural language documents such as prescription, letters, surgery reports, etc. All documents are written in everyday language.

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]. Then, a semantic annotator based on Natural Language Processing (NLP) called *Multi-Terminological Concept Extractor* (ECMT) [2]. Finally, a semantic multilingual search engine [3].

To improve the ECMT, a new strategy using deep learning techniques was defined. To implement it, a new text representation had to be designed to fit the input of neural networks algorithms.

Word embedding

In NLP, a chosen representation has to keep the semantic similarities between different words from a corpus of texts. Thus, the representation of a unique token has to show its proximity to other related meaning concepts, as illustrated in the quotation "*You shall know a word by the company it keeps"* [4], now known as the *distributional hypothesis*.

In fact, a compact and precise representation of words could bring several benefits. First, computers are way better to perform operations on low-dimensional objects. Second, probabilities calculation or mathematical operations can be done on words, such as the famous "*(king – man) + woman ~ queen*". And finally, 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.

Implementations

Word2Vec

The word2vec approach was the first modern embedding released in 2013 [5]. Mikolov *et al.* implemented two kinds of architectures. The Continuous Bag of Word (CBOW) architecture treats the entire context as a single observation. A hierarchical softmax was also used to reduce computational limits [6]. The input layer accepts one-hot encoding as input (a sentence is encoded as a very sparse vector composed of 0 or 1, depending on the words found in this sentence). The Skip-Gram (SG) architecture uses a sliding window to define "*context / target*" pairs (*e.g.*, "*How / you*" is the context of the word "*are"* in the sentence "*how are you?*"). The entire corpus **V** will thus be transformed into many pairs *context / target* (*i.e., input / output* of the network). To reduce the computation of such an amount of data (in a "normal" training situation, all the weights should be updated in each passing through an example), the authors brought some new tricks. Word pairs appearing always together are treated as single tokens, frequent words subsampling and negative sampling [7].

GloVe

This model was released by the Stanford University [8]. 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 of shape **V×C** with **V** being the vocabulary of the corpus and **C** context examples. The probability ($Vw_1 \| Vw_2$) of a word Vw_1 being close to another Vw_2 will increase during the training. This gigantic matrix is then factorized by using the log function.

FastText

FastText is a newly released model and comes from a new idea [9]. Bojanowski *et al.* consider that a word could be the result of all of the vector decomposition of this word (sub word model). Each word V_w can be decomposed into a set of *n*-grams vectors. For example, the word "*boat*" can be seen as a set of *n*-gram with $n = 3$ as [*b* + *bo* + *boa* + *o* + *oa* + *oat* + *a* + *at* + *t*]. Thus, each word is embedded in the vector space as the sum

of all vectors composing this token, incorporating morphological information into the representation. Like Word2Vec, FastText comes with the two different architectures (SG and CBOW).

Related work

For a few years, the huge interest in word embedding led to comparison studies. Scheepers, Gavves, and Kanoulas [10] compared the three word embedding methods presented here but the three models were trained on different datasets (Word2Vec on news data, while FastText and GloVe trained on more definitional data, Wikipedia and Common Crawl respectively). Bairong *et al.* [11] also performed a comparison between these three, but focused on bilingual automatic translation comparison (BLEU score [12]) and without human evaluation for all the different models. More recently, Beam *et al.* [13] produced huge publicly available word embeddings based on medical data, however they didn't compare FastText, only Word2Vec and GloVe. Finally, Wang *et al.* [14] compared word embedding training set influence on models utilization, and its impact on different NLP tasks related to medical applications.

Moreover, many different teams or companies have released pre-trained word embedding models (*e.g.,* Google, Stanford University, etc.). However, in a clinical context, the vocabulary coverage of those embeddings is quite low regarding the words used. Indeed, many misspells, acronyms of specific abbreviations are regularly found in the documents produced in the hospital. Thus, a local training on specific data is often needed, especially with languages other than English.

Contributions

The objective here is to compare these five different methods to obtain the best possible words embedding (Word2Vec SG and CBOW, GloVe, FasText SG and CBOW). 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. The influence of the number of documents in the training set will also be assessed.

Methods

Corpus

The corpus used in this study is composed of health documents from the SHDW of the RUH, France. All these documents are in French. They are also quite heterogeneous but their type is stored in the SHDW: 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.

Document de-identification

These documents were then de-identified to protect each identity of every patient or doctor from the RUH. Every of the 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*>.

Pre-processing

Texts have been split into token lists, the data has been lowered (meaningless to make distinction), the punctuation was removed, and the numerical values were replaced by a meta-token <number>. We chose to not remove stop words, due to their negligible impact on the context. Indeed, their multiple apparitions in many different contexts will just create a cluster of stop words in the middle of the VSM.

Training

Models 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 the models' hyper-parameters on the literature and on our own experience [15]. All chosen values are listed on Table 1. The *minimum count* parameter was set higher than usual settings due to the large quantity of data in the training set.

Table 1— Hyperparameters Used to Train the Five Word Embedding Models

Parameter	Model	Value
epochs	All	100
Min. count	All	20
Window size	All	7
Learning rate	All	2.5×10^{-2}
Embedding size	All	80
Alpha rate	All	0.05
Negative sampling	Word2vec/FastText	12
Subsampling	GloVe	$1e^{-6}$

Evaluation

Cosine similarity

We compare how similar the embeddings for a pair of concepts are by computing the cosine similarity of their corresponding vectors, and then using this similarity to assess whether or not the two concepts are related.

We used two well-known validations set UMNSRS-Similarity and UMNSRS-Relatedness, containing 566 and 588 manually rated pairs of concepts respectively [16]. However, since our corpus was in French, we used the translated and aligned version of the MeSH (Medical Sub-Heading) terminology stored in the ECMT to translate these two sets. The result provides a number of 308 pairs for the UMNSRS-Sim and 317 for the UMNSRS-Rel.

Mathematical operations

Mikolov's paper presenting Word2Vec showed that mathematical operation on vectors such as additions or subtractions are possible. Mathematical operations covering a wide range of possible subjects found in the selected documents (hospital departments, human tissues, biology, drugs) were defined.

Odd one out

The odd one out similarity task tries to measure the model's accuracy 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 cluster with the two others [17]. To create such a validation corpus, 53 pairs of concepts potentially linked in a medical text have been defined from the MeSH terminology. Then, 53 words appearing more than 1,000 times in the corpus have been randomly selected to be used as odd terms.

Human evaluation

A formal evaluation of the five methods was performed by a medical doctor (SJD). A list of 112 terms has been extracted from the Medical Sub Heading (MeSH) terminology, covering every possible branch. SJD then assessed the relevance of the top five closest word vectors returned for each of the 112 concepts by the five created models. Retrieved citations were assessed for relevance according to a three-point modality scale

used in other standard information retrieval test sets: bad (0), partial (1) or full relevance (2).

Training set influence

To go further, models are going to be trained twice. First by randomly selecting 5% of the total amount of available documents in the RUH (~600K) and all evaluation tasks were performed against the five models. Then, the entire corpus (~12M) was used as a training set, and those evaluation tasks were assigned to these newly trained models.

Results

Corpus

In total, 641,279 documents for the first phase and 11,762,100 for the second one had been extracted from the RUH. They had been de-identified and pre-processed. Regarding the vocabulary, 355,597 unique tokens are found in those 12M of documents. However, this number can be pondered with 170,433 words appearing less than 5 times 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 (*sd* = 207.42).
These documents were decomposed using the Term-Frequency Inverse-Document-Frequency (TF-IDF) algorithm, results in a frequency matrix. Vectors have been used to clusterize those documents with a kMeans algorithm (number of classes **K = 5**). To visualize their distribution on two dimensions, the t-SNE algorithm has been used (figure 1).

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, the vocabulary itself contained in the documents from the HDW being 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 kind of contexts (short sentences, acronyms and abbreviations, highly technical vocabulary...). Regarding drug prescriptions and letters to a colleague or from a general practitioner, they present more specific vocabulary (drugs and chemicals, and current/sustained 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 min to process the entire corpus of 600K documents (Table 2).

Table 2— Algorithms Training Time (minutes) Regarding the Number of Documents

All time are given in minutes. GloVe is the fastest algorithm to train.

Algorithm	600K documents	12M documents
Word2Vec SG	182.0	497.9
Word2Vec CBOW	33.4	308.8
GloVe	**17.5**	**65.9**
FastText SG	1678.1	5573.8
FastText CBOW	1577.0	4974.0

GloVe performs much better in terms of computational time due to the way it handles the vocabulary. GloVe 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 word sub-vectors it creates. However, for Word2Vec, the difference between the two available sub-architectures is highly significant (33 min to 3h02 for 600K documents). 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 percentages of validated pairs from the UMNSRS datasets are presented in Table 3. FastText SG performed this task with the highest score (3.89% and 5.04% of valid pairs found for UMNSRS-Sim and UMNSRS-Rel respectively with 600K documents, 6.43% and 7.13% with 12M documents). The very low scores indicate that this kind of published dataset is useful to validate models trained on more academic texts, not those written in natural language. Some words will never be found because of the use of an acronym by health practitioners ("*HTA*" instead of "*HyperTension Artérielle*") or because of the informal form, mainly used in these kinds of documents.

Table 3— Percentage of Pairs Validated by the Five Trained Models on Two UMNSRS Evaluation Sets

	600K documents		12M documents	
	U-Sim	U-Rel	U-Sim	U-Rel
W_SG	2.92	4.10	4.73	6.92
W_CBOW	3.57	4.10	5.12	6.92
GloVe	1.29	0.94	1.35	0.94
F_SG	**3.89**	**5.04**	**6.43**	**7.13**
F_CBOW	**3.89**	3.79	**6.43**	6.65

Mathematical operations

A list of six mathematical operations has been defined with the help of a medical doctor and a university pharmacist (listed in Table 5). Each operation consists in verifying if *(term_1 – term_2) + term_3 ~ term_4* is true.

Table 4— Logical Operations on Words Having to be Retrieved with the Different Trained Models

Each is listed in English but has been performed against models in French.

1	(cardiology - heart) + lung ~ pneumology
2	(melanoma - skin) + gland ~ adenoma
3	(corpuscle - blood) + immune ~ immunoglobulin
4	(furosemide - kidney) + heart ~ fosinopril
5	(limb - lower) + upper ~ arm
6	(morphine - opioid) + antalgic ~ perfalgan

The number of validated operations performed by each model is presented on Table 4. Word2Vec SG gets the highest score on this task (5/6, regardless of the number of documents in the training set), while GloVe gets 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.

Table 5— Score for Mathematical Operation Tasks on Six Point Maximum for Each of the Five Trained Models

Algorithm	600K documents	12M documents
Word2Vec SG	**5**	**5**
Word2Vec CBOW	3	3
GloVe	2	3
FastText SG	3	4
FastText CBOW	3	3

Odd one similarity

Regarding the odd one similarity task, Word2Vec SG is the best so far with 65.38% (600K documents) and 75.5% (12M documents) of odd-one terms correctly found (Table 6).
Regarding the sub-architectures presented by both Word2Vec and FastText, the SG always performed better than the CBOW, possibly due to the negative sampling. In fact, the studied corpus is quite heterogeneous, and a word can be listed as items (*e.g.,* drugs in prescriptions) instead of being used in correct sentences. Sometimes, the complete update of vectors' dimensions generates non-sense in the models.

Table 6— Percentage of Odd One Tasks Performed by Each of the Five Trained Models

Algorithm	600K documents	12M documents
Word2Vec SG	**65.4**	**75.5**
Word2Vec CBOW	63.5	69.8
GloVe	18.5	39.6
FastText SG	44.4	45.8
FastText CBOW	40.7	41.3

Human Evaluation

The evaluation focused on 2800 terms (5 vectors × 112 MeSH concepts × 5 models), and was performed by two evaluators, CM and SJD, on models trained with 600K and 12M documents. First, the accordance between CM and SJD was assessed with a weighted kappa test (k = 0.6133). According to the literature, the agreement between the two evaluators can be considered as substantial [17]. 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.
Moreover, to assess if human evaluators remained consistent regarding the cosine score computed by each model, we compared the average note given by the two evaluators with the average of the cosine distance computed for each model (table 7). Word2Vec with the SG architecture performed the highest score, regardless of the evaluator (1.469 with 600K documents and 1.420 with 12M). Interestingly, GloVe computes the shortest cosine distance in averages (0.884 on the 112 given concepts), while both evaluators gave it the lowest grade.

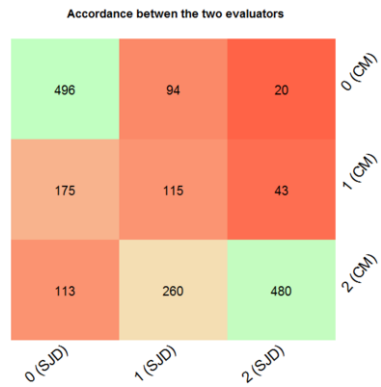

Figure 2— Global Representation of the Notation Accordance between the Two Evaluators (CM and SJD).

Notes attributed to a model output are going from 0 (bad matching) to 2 (good matching). Colors are ranging from light green (very similar) to red (completely different).

Table 7— Comparison between Cosine Distance Computed by each Model and the Human Evaluation Performed

Notes and distances are in averages on the top-5 closest vectors for 112 queries on every model.

Algorithm	600K documents		12M documents	
	Cos	Eval	Cos	Eval
Word2Vec SG	0.731	**1.469**	0.785	**1.420**
Word2Vec CBOW	0.776	1.215	0.716	1.33
GloVe	**0.884**	0.703	0.692	1.27
FastText SG	0.728	1.156	**0.963**	0.25
FastText CBOW	0.748	1.131	0.930	0.466

Discussion

In this study, the three most famous word embeddings have been compared. Word2Vec SG got the best score for three out of the four rated tasks (FastText SG is the best regarding the cosine one). These results are coherent with those obtained by Muneeb *et al.*, which compared Word2Vec and GloVe with the cosine similarity task [19]. GloVe had the worst grade, however it's the fastest to train so far. Regarding FastText, it is interest-

ing to note that the morphosyntactic similarities are kept in account in the vector space creation. Moreover, the sub-vector decomposition of words allows this kind of model to be queried by words absent from the original training corpus. We can imagine this model being used for orthographic correction or acronym disambiguation.
Interestingly, the size of the training set does not heavily influence Word2Vec (Table 7), but GloVe seems to improve its quality according to the human annotators. In fact, designed with a count-based method, this algorithm is directly affected by the amount of available data to train. FastText is highly degraded with more documents. The relevance of the returning vectors remained low because of the high proportion of morphosyntactic similarities between the sent and the returned tokens.
The corpus used as a training set comes from a real work environment. Finding a good evaluation for embedding produced in such a context is a hard task, and the performances shown by some models trained on scientific literature are often biased.
Future work will assess whether the best embedding method could help for semantic concept enrichment.

Conclusion

In our case, Word2Vec with the SG architecture got the best grade regarding three of the four rated tasks.
Any end user is now able to query the word embedding models produced on a dedicated web site as well as to download high quality dimension reduction images and test sets (URL: https://cispro.chu-rouen.fr/winter/) .

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 thanks Catherine Letord, pharmacist, Jean-Philippe Leroy, M.D. for their help in creating the test datasets.

References

[1] J. Grosjean, T. Merabti, B. Dahamna, I. Kergourlay, B. Thirion, L.F. Soualmia, and S.J. Darmoni, Health multi-terminology portal: a semantic added-value for patient safety, *Studies in Health Technology and Informatics*. **166** (2011) 129–138.
[2] S. Sakji, Q. Gicquel, S. Pereira, I. Kergourlay, D. Proux, S.J. Darmoni and M.H. Metzger, Evaluation of a French medical multi-terminology indexer for the manual annotation of natural language medical reports of healthcare-associated infections., *Studies in Health Technology and Informatics*. **160** (2010).
[3] R. Lelong, L. Soualmia, B. Dahamna, N. Griffon, and S.J. Darmoni, Querying EHRs with a Semantic and Entity-Oriented Query Language, *Studies in Health Technology and Informatics*. **235** (2017) 121–125.
[4] J.R. Firth, A Synopsis of Linguistic Theory, *Basil Blackwell Oxford*. **59** (1957) 168–205.
[5] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean, Distributed Representations of Words and Phrases and their Compositionality, *NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems*. **2** (2013) 3111–3119. doi:arXiv:1310.4546.
[6] A. Mnih, and G. Hinton, A Scalable Hierarchical Distributed Language Model, *Proceedings of the 21st International Conference on Neural Information Processing Systems*. **21** (2008) 1081–1088.
[7] M. Baroni, G. Dinu, and G. Kruszewski, Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. **1** (2014) 238–247. doi:10.3115/v1/P14-1023.
[8] J. Pennington, R. Socher, and C.D. Manning, GloVe: Global Vectors for Word Representation, *Conference: Conference: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. **14** (2014) 1532–1543. doi:10.3115/v1/D14-1162.
[9] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, Enriching Word Vectors with Subword Information, *Transactions of the Association of Computational Linguistics*. **5** (2017) 135–146. doi:arXiv:1607.04606.
[10] T. Scheepers, E. Gavves, and E. Kanoulas, Analyzing the compositional properties of word embeddings, *University of Amsterdam*. (2017).
[11] Z. Bairong, W. Wenbo, L. Zhiyu, Z. Chonghui, and T. Shinozaki, Comparative Analysis of Word Embedding Methods for DSTC6 End-to-End Conversation Modeling Track, *Tokyo Institute of Technology*. (2016). doi:arXiv:1706.07440.
[12] K. Papineni, S. Roukos, T. Ward, and W.J. Zhu, BLEU: a method for automatic evaluation of machine translation, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. **40** (2001) 311–318. doi:10.3115/1073083.1073135.
[13] A.L. Beam, B. Kompa, I. Fried, N.P. Palmer, X. Shi, T. Cai, and I.S. Kohane, Clinical Concept Embeddings Learned from Massive Sources of Multimodal Medical Data, *ArXiv*. (2018). doi:arXiv:1804.01486.
[14] Y. Wang, S. Liu, N. Afzal, M. Rastegar-Mojarad, L. Wang, F. Shen, and H. Liu, A Comparison of Word Embeddings for the Biomedical Natural Language Processing, *Journal of Biomedical Informatics*. **85** (2018).
[15] B. Chiu, G. Crichton, A. Korhonen, and S. Pyysalo, How to train good word embeddings for biomedical NLP, *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*. **15** (2016) 166–174. doi:10.18653/v1/W16-2922.
[16] S. Pakhomov, B. McInnes, T. Adam, Y. Liu, T. Pedersen, and G.B. Melton, Semantic similarity and relatedness between clinical terms: an experimental study, *AMIA Annual Symposium Proceedings*. **2010** (2010) 572–576.
[17] J. Sinapov, and A. Stoytchev, The odd one out task: Toward an intelligence test for robots, *IEEE 9th International Conference on Development and Learning*. **9** (2010). doi:10.1109/DEVLRN.2010.5578855.
[18] M.L. McHugh, Interrater reliability: the kappa statistic, *Biochemia Medica*. **22** (2012) 276–282.
[19] M. Th, S. Sahu, and A. Anand, Evaluating distributed word representations for capturing semantics of biomedical concepts, *Proceedings of BioNLP 15*. **15** (2015) 158–163. doi:10.18653/v1/W15-3820.

Address for correspondence

Emeric Dynomant
emeric.dynomant@omictools.com
+33 6 59 90 12 49