e-Health – For Continuity of Care
C. Lovis et al. (Eds.)

doi:10.3233/978-1-61499-432-9-584

Exploring the Application of Deep Learning Techniques on Medical Text Corpora

José Antonio MINARRO-GIMÉNEZ [a], Oscar Marín-Alonso [a,b] and Matthias SAMWALD [a,1]
[a] *Section for Medical Expert and Knowledge-Based Systems, Medical University of Vienna, Vienna, Austria*
[b] *Dept. of Computer Technology, University of Alicante, Alicante, Spain*

Abstract. With the rapidly growing amount of biomedical literature it becomes increasingly difficult to find relevant information quickly and reliably. In this study we applied the word2vec deep learning toolkit to medical corpora to test its potential for improving the accessibility of medical knowledge. We evaluated the efficiency of word2vec in identifying properties of pharmaceuticals based on mid-sized, unstructured medical text corpora without any additional background knowledge. Properties included relationships to diseases ('may treat') or physiological processes ('has physiological effect'). We evaluated the relationships identified by word2vec through comparison with the National Drug File – Reference Terminology (NDF-RT) ontology. The results of our first evaluation were mixed, but helped us identify further avenues for employing deep learning technologies in medical information retrieval, as well as using them to complement curated knowledge captured in ontologies and taxonomies.

Keywords. Deep learning, biomedical literature, evaluation system

Introduction

The large amount of biomedical information in databases such as PubMed [1] is a valuable source for information extraction [2]. Information extraction and, in particular, natural language processing methods are required to annotate and process biomedical literature [3].

The concept of 'deep learning' has recently gained a lot of attention. It refers to unsupervised learning algorithms which automatically discover data without the need of supplying specific domain knowledge [4]. This approach usually has higher performance rates than supervised and informed methods when processing large unstructured corpora. However, the ability of these algorithms has to be evaluated to measure their error rate.

Word2vec[2] implements an efficient deep learning algorithm for computing high-dimensional vector representations of words and their relationships [5], based on unstructured text data. Namely, Word2vec provides two architectures, continuous bag-

[1] Corresponding Author.
[2] https://code.google.com/p/word2vec/

of-words and skip-gram, for computing continuous distributed vector representation of words from large datasets (up to hundreds of billions of words). Therefore, Word2vec requires training the corpora using one of such architectures and setting some parameters, such as the dimensionality of the vector space or the size of the context window. Word2vec provides two tools to exploit trained corpora: *Distance* and *Analogy*. The *Distance* tool retrieves a vector of words which are closely related to a given word in the corpora. The results also contain the corresponding cosine distances of each word that indicates how close it is related to a given word. The *Analogy* tool, on the other hand, is able to detect linguistic regularities in a vector representation [6], e.g. the analogy tool aims to search for the information related to the relationship *is_capital_of* with an example of such relation in the text, *France* - *Paris*, and a particular city capital, *Berlin*, and resulting in a vector similar to the word *Germany*. Finally, these tools can be used to build more complex natural language processing and machine learning applications.

In this paper, we evaluate the word2vec toolkit on mid-sized medical text corpora. We compared the results of word2vec with knowledge encoded in the National Drug File – Reference Terminology (NDF-RT) ontology[3] to validate its results.

1. Methods

The evaluation was carried out in four steps: (1) gathering and processing openly available medical text corpora; (2) training vector space representations of these corpora with word2vec; (3) test word2vec tools and compare their results against the reference ontology; (4) calculating result statistics.

We created a script for crawling medical data from online repositories using PHPcrawl[4] and API calls. We gathered clinically relevant medical information from PubMed, Merck Manuals[5], Medscape[6] and Wikipedia[7]. We created scripts for stripping non-relevant portions of web pages (such as headers and footers) and HTML markup that would degrade the quality of the corpora. Table 1 gives an overview of the medical corpora created through this process.

Table 1. Corpora used in the experiment. Word counts refer to the final corpora that were derived from source datasets after all processing steps. Vocabulary sizes refer to the number of distinct words found in each corpus.

Corpus	Word count	Vocabulary size
Clinically relevant subset of PubMed, full abstracts	161.428.286	204.096
Conclusion sections from clinically relevant subset of PubMed, *"pubmed_key_assertions"*	17.342.158	47.703
Merck Manuals	12.667.064	49.174
Medscape	25.854.998	63.600
Clinically relevant subset of Wikipedia, *"wikipedia"*	10.945.677	65.875
Combined corpus (including all corpora above), *"combined"*	236.835.672	261.353

These corpora needed to be pre-processed before training since word2vec has no built-in functionalities for term normalisation or dealing with punctuation. We found

[3] http://bioportal.bioontology.org/ontologies/NDFRT
[4] http://astellar.com/php-crawler/
[5] http://www.merckmanuals.com/
[6] www.medscape.com/
[7] http://www.wikipedia.org/

that raw corpora contained an abundance of syntactic variations that had a negative impact on trained corpora and, therefore, the quality of the resulting vector space models. We identified the need of removing all punctuation, transforming the corpora to lower-case, and forming multi-word terms due to the fact that word2vec indexes only single words from the corpora.

Once the gathered corpora were processed to improve their content, we used the training tool of word2vec to generate the corresponding vector model. The corpora were trained using the parameters indicated in the Table 2 to obtain their vector-models.

Table 2. This table provides an example of the parameters used with the word2vec training tool to train the medical corpora. In this example the training tool uses the corpus file "corpora.txt" to generate the vector model into the file "vector-model.bin". It uses the skip-gram architecture with a vector size of 200 and a window of 5. Besides, the vector model was generated hierarchical softmax training algorithm, a threshold of 1e-3 for downsampling the frequent words. The training tool uses 12 threads execution to generate the binary file of the vector model.

Command line execution
word2vec *-train* corpora.txt *-output* vector-model.bin *-cbow* 0 *-size* 200 *-window* 5 *-negative* 0 *-hs* 1 *-sample* 1e-3 *-threads* 12 *-binary* 1

The evaluation of the trained corpora was focused on testing the results of the word2vec distance and analogy tools. The resulting vectors consisted of the 40 most closely related terms in the text for each queried term, determined by cosine distance in the vector space model. A good (i.e., high) cosine distance value meant that two terms were determined to be related by the word2vec algorithm. To compare the resulting vectors we use the manually curated content of the NDF-RT ontology described in Table 3.

Table 3. This table describes the information associated to each relationship of NDF-RT ontology that was collected to do the evaluation of word2vec tool-kit and to form multi-word terms during the processing stage of the pre-trained corpora. Besides, the domain and range of each relationship is described.

NDF-RT relationship	Description
may_treat	Provides the association between drugs and the diseases they may treat.
may_prevent	Provides the list of diseases that a drug may prevent.
has_PE	Relates drugs to their corresponding physiological effects.
has_MoA	The mechanisms of action of each drug.

To perform this evaluation, we developed a system that automatically queried one trained corpus using the analogy and distance tools and matched the resulting vectors with the content of the NDF-RT ontology. Figure 1 shows the software architecture of this evaluation system.

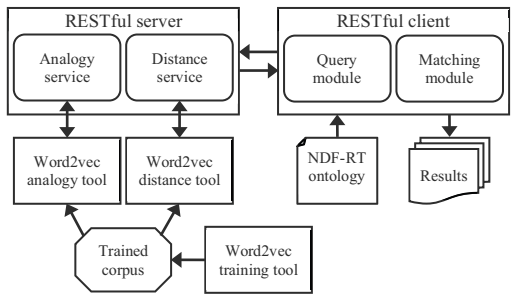

Figure 1. Evaluation system architecture defined to execute word2vec tools and compare their results.

We obtained the following statistics: (i) we checked the number of resulting vectors with at least one correct term from the relationships of NDF-RT; (ii) we computed the distribution of each correct term in the distance vector; and (iii) we

obtained the hit rate of word2vec for a selected subset of terms, for which we determined if they appeared in close proximity in the used corpora.

2. Results

Most of the software we developed for running these evaluations is available open-source on Google Code[8].

We obtained three processed medical corpora and three vector models – "pubmed", "wikipedia" and "combined" – which can be further exploited with word2vec tools. Besides, we developed a system that publishes the functionality of distance and analogy tools using RESTful services, and automatically collect the resulting vectors to evaluate their content with external information sources.

Table 4 provides statistics on the hit rates of correct terms in the vectors retrieved by word2vec tools. The hit rate represents the percentage result lists with at least one correct term that matches the data from the ontology. On the one hand, the best hit rate was yielded using the analogy tool for retrieving "may_treat" information from the "combined" corpus, with a hit rate of 27,37%. On the other hand, the worst hit rate was 0,32% for the result of the "wikipedia" corpus with the distance tool and the "has_PE" relationship.

Table 4. Hit rates obtained by comparing the content of the vectors retrieved by distance or analogy tools, with the information of "may_treat", "may_prevent", "has_PE" and "has_MoA" from NDF-RT ontology.

Corpora	may_treat		may_prevent		has_PE		has_MoA	
	Analogy	*Distance*	*Analogy*	*Distance*	*Analogy*	*Distance*	*Analogy*	*Distance*
combined	27,37%	3,21%	10,59%	6,32%	2,49%	0,67%	6,91%	6,81%
pubmed	15,74%	2,13%	5,09%	4,07%	0,84%	0,37%	1,51%	3,60%
wikipedia	14,9%	1,3%	5,35%	3.34%	2,22%	0,32%	2,69%	3,34%

Figure 2 shows the second statistics about the distribution of positions of correct terms in the result list. In our evaluation the mean position of correct terms in the result list is close to 16, with a standard deviation higher than ten. Besides, the low probability of the last terms in result lists indicates that it is unlikely to find more correct terms in a result list by increasing the length of the result list.

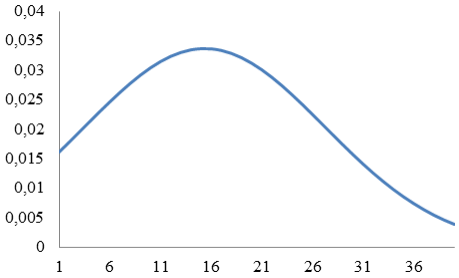

Figure 2. Distribution of the positions of correct words in the vectors as a result of analogy tool with the "combined" corpus and testing "may_treat" relationship.

Table 5 displays the hit rates of 15 drugs when querying "may_treat" information with the analogy tool and the "combined" corpus. We checked how many of the correct terms were in proximity to a particular drug term within a window size of 20 words

[8] https://code.google.com/p/biomedical-text-exploring-tools/

anywhere in the corpus. Then, we calculated how many of those words were found by word2vec tools. We found that the hit rate of the word2vec tool to be approximately 51% and the hit rate in a 20-term window to be approximately 80%.

Table 5. Comparison of the hit rates of 15 drugs using the analogy tool and a simple proximity text search with a 20-term window (for the "combined" corpus and the "may_treat" relationship). A value of 100% means that all of the diseases in a "may_treat" relationship with the drug were recovered.

Drug term	20-term window hit rate	word2vec hit rate
Enoxaparin	40%	33,33%
Fluvastatin	100%	50%
Diazepam	75%	75%
Valporic_acid	66,67%	66,67%
Omeprazole	50%	28,57%
Trientine	100%	50%
Natamycin	100%	7,69%
Valsartan	75%	75%
Prochlorperazine	100%	40%
Formoterol	100%	50%
Imatinib	100%	66,67%
Chloroquine	25%	16,67%
Atorvastatin	100%	75%
Perindopril	75%	75%
Gemfribrozil	100%	66,67%
Mean	80,44%	51,75%

3. Discussion

The obtained statistics showed a low hit rate of word2vec tools. Consequently, the ability of word2vec to retrieve significant words from the restricted corpora we used is not suitable for applications requiring high precision.

Further analyses need to be conducted to test if different parameters of the training tool, such as window size, vector size, or the bag-of-words architecture have a significant impact in the efficiency of the word2vec tools in extracting relevant information.

References

[1] Roberts RJ. PubMed Central: The GenBank of the published literature. *Proc Natl Acad Sci USA.* 2001;16: 381-382.

[2] Kim JD, Ohta T, Tsujii J. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*. 2008; 9: 10.

[3] Albright A, Lanfranchi A, Fredriksen A, Styler WF, Warner C, Hwang JD, et al. Towards comprehensive syntactic and semantic annotations of the clinical narrative. *J Am Med Inform Assoc.* 2013; 20: 922-930.

[4] Bengio Y, Courville A, Vincent P. Representation Learning: A Review and New Perspectives. *IEEE Trans*. PAMI, special issue Learning Deep Architectures. 2013 Aug; 8(35): 1798 – 1828.

[5] Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at International Conference on Learning Representations; 2013 May 2-4; Scottsdale: Arizona.

[6] Mikolov T, Yih W, Zweig G. Linguistic Regularities in Continuous Space Word Representations. In Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2013 June 9-15; Atlanta: Georgia.