Experimental Results (2022), 1–6
doi:10.1017/exp.2022.xx

COMPUTER SCIENCE
REPLICATION-RESULT

Semantic similarity detection in medical field based on convolutional neural network

Tianrui Liu[1]

[1]The Grainger College of Engineering, University of Illinois at Urbana-Champaign, Champaign, IL, 61820, USA., Corresponding author. E-mail: tl49@illinois.edu

(Received xx xx 2022; Revised xx xx 2022; Accepted xx xx 2022)

Abstract

The purpose of machine learning in computer science is to make machines more efficient and reliable. In healthcare, machines are an extension of the doctor's brain and a force multiplier. After all, patients always need human touch and care, which machines cannot provide. Therefore, the job of machines is not to replace doctors but to help them provide better service and care. So in the process of identifying electronic medical records using the machine not only improves the efficiency of the doctor's treatment but avoiding errors in the manual review. Accordingly, we improve the precision and accuracy of medical terminology similarity computation with a large dataset of medical questions and answers using convolutional neural networks. We reproduce and improve it through a method in which we will document the shortcomings and advantages and present our discussion and evaluation of this method. We will make available the complete dataset and the algorithm model code.

Keywords: word embedding; convolutional neural network; semantic similarity algorithm; natural language processing

Introduction

With the emergence of massive medical data, much biomedical information is available. For example, information on diseases, genes, and drugs is presented in the scientific literature in an unstructured manner (Fernández-Suárez et al., 2014). As mentioned above, unstructured medical information is not only effective in helping to achieve knowledge discovery in the field of medicine (Vishrawas et al., 2019). It also plays a crucial role in healthcare organizations. However, the fact is that there is extreme complexity and variability in these unstructured data, such as imaging examination techniques. We plan to promote and improve a machine learning-based approach to more effectively retrieve these patients from EMR. (Tao et al., 2019). Since few studies have fused machine learning for semantic representation techniques and topic-level knowledge graphs to explore similarity computation methods for medical texts. Furthermore, Few computations are based on similarity-based subject words. So we plan to use similarity computation models in machine learning and convolutional neural networks to analyze the similarity of medical texts.

Research on medical text similarity computation focuses on obtaining the similarity between sentences by calculating word-level similarity, then used for knowledge discovery in the medical field. Convolutional neural networks are effective in capturing the semantic similarity between texts. Thus, it can perform well in text-matching tasks (Aliaksei Severyn and Alessandro Moschitti, 2015; Yin et al., 2016).

In this study, we were designed to help physicians and clinical quality control personnel efficiently retrieve patient examination reports. Moreover, it is improved and enhanced based on an end-to-end solution based on convolutional neural networks proposed in this paper (Tao et al., 2019). Comparative trials are also conducted with traditional methods such as keyword mapping, latent semantic analysis (Deerwester et al., 1990), latent Dirichlet allocation (Blei et al., 2003), Doc2Vec (Quoc Le and Tomas Mikolov, 2014), Siamese LSTM (Mueller et al., 2016) and named entity recognition (Yonghui et al., 2015).

Objective

In this paper (Tao et al., 2019), a convolutional neural network-based model is proposed. It identifies reports of imaging examinations and pathological examinations that contain repeated body parts by detecting semantic similarities. The structure of the model can be divided into an input layer, a feature extraction layer, and a fully connected layer. A new feature vector is obtained by mapping the vector through the input layer and initializing it. It is passed to the fully connected layer and the output layer to calculate the likelihood of containing duplicate body parts. The graph embedding method is then used to initialize the third vector. Medical concept vectors are trained by semantic relational information of medical ontologies (Aditya Grover and Jure Leskovec, 2016; Tomas et al., 2013). Furthermore, the CNN models with both random initialization and pre-trained word vectors outperformed any other baseline models in terms of AUC scores and improved about 3%-7%.

Since the original paper does not have open-source code for the authors' experimental model, we expect to modify and train according to the other open-source code provided in the paper methodology. Furthermore, evaluate the same environment for the same data. A comparison of the two models before and after the modification is made to confirm whether the modified model performs better than the original model in the experiment.

We plan to test the following claims:

- The AUC score of the CNN model using medical concept vectors is 0.8% higher than that of the model using random initialization vectors. The precision of the CNN model using medical concept vectors is 1.9% higher than that of the model using random initialization vectors.
- The accuracy of similarity of our CNN model is 5% higher than the accuracy of similarity of the Doc2Vec model, which is one of the baseline models mentioned in the paper.
- Some medical terms that are not well separated have some separation errors that prevent accurate analysis of pathology reports.

Methodology

Because the authors did not open source the source code, we plan to seek other relevant code on the Internet and follow the specifications below to develop our model and evaluation. Furthermore, the original article focused mainly on Chinese case studies. It did not focus excessively on English medical terminology analysis. We plan to extend this convolutional neural network to the English domain.

Model and Data Descriptions

For the Chinese word separation component segmentation, we plan to use an open-source tool pkuseg (Ruixuan et al., 2019) developed by Peking University. We choose this toolkit because it has a pre-trained model that focuses on the medical domain. With pkuseg, we can separate medical words from other ordinary words. After the segmentation, we filter out words that are not in the medical domain by using the medical vocabulary dictionary THUOCL [1] developed by Tsinghua University.

For the analysis of Chinese word vectors, as we said, we plan to use two kinds of vectors: one is a random initialization vector, and the other is a medical concept vector. Since we do not have time, data, and computational power to train a new model, we plan to use Chinese-Word2vec-Medicine [2] for the medical

[1] https://github.com/thunlp/THUOCL

[2] https://github.com/WENGSYX/Chinese-Word2vec-Medicine

concept vector. As for the randomly initialized word vector, since the original paper does not mention how to build it, we adopt the same vocabulary set and vector size. Moreover, we assign each word in vocabulary a randomly generated vector.

About our explanatory machine learning classifier for prediction of LIME experimental code source Github's open-source project [3]. We used matplotlib to visualize two randomly selected medical reports containing duplicates in the test set. LIME model was used for processing.

Our data comes from two parts. The first is the Chinese medical dataset. Next is the dataset about the English language.

- Dataset for Chinese community medical question answering: Since the data this paper used is not public, and we cannot find other substitutes, the dataset for Chinese community medical question answering is the only available and suitable data we can use. This dataset contains questions from patients and answers from doctors, and we mainly use the answer part, as the question part contains too much unrelated and unprofessional words [4].
- For Electronic Health Records in English, our data were obtained from PhysioNet's MIMIC database. We mainly refer to natural language questions. As with the Chinese dataset, the existing QA dataset on the English EHR uses unstructured clinical records to retrieve answers. We can use a similar approach to solve the data problem [5].

Hyperparameters

- For Doc2Vec, the baseline model to be tested, we set the vector size to its default size **100** and vocabulary size to our training corpus vocabulary size. Since we have little experience tuning Word2Vec, we feel that using default parameters is the safest option.
- For our CNN models, we follow the instruction in the paper. To eliminate the effect of non-related variables in the experiment, the vector size of pre-train medical word embedding is used, and it is **512**. According to the paper, there are three different sizes of the kernel (filters): **3**, **4**, **5**. Given that our word vector size is **512**, the kernel shapes are $(\mathbf{3}, \mathbf{512})$, $(\mathbf{4}, \mathbf{512})$, $(\mathbf{5}, \mathbf{512})$. Also, we have **32** filters for each size. Thus, we have a total of **96** convolution filters. Then we apply **MaxPool1d** on each filtered result. And finally, a fully connected layer with input size **96** and output size **1**. For **learning rate**, we use **0.01**, which will yield higher accuracy. And a batch size of **32**. The number of epochs we use is **4**.

Implementation

Our data overlap amount is identified as positive or negative by Alibaba training. confirm and define the threshold value (original overlap amount is not public. (By body part if the overlap is positive, otherwise negative).

- We use pkuseg as a Chinese medical word separation tool. This is because pkuseg can achieve higher word separation accuracy. It also supports self-training models, so we can test the models better and improve the accuracy [6].
- Chinese-Word2vec-Medicine is the baseline test implemented in our experiments. It is the only sizeable open-source word vector tool for Chinese in the medical domain. We compare the word vectors by training corpus [7].

[3] https://github.com/marcotcr/lime

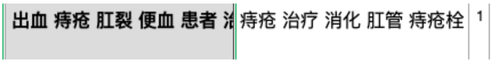
[4] https://github.com/zhangsheng93/cMedQA2

[5] https://physionet.org/content/mimic-iii-question-answer/1.0.0/

[6] https://github.com/lancopku/pkuseg-python

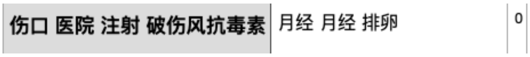
[7] https://github.com/WENGSYX/Chinese-Word2vec-Medicine

- Since we are not physicians, we use a pre-train model to help us decide whether two sentences share the same body parts. We planned to use a pre-train model from Alibaba (Ningyu et al., 2020) [8] at first, but it is difficult to use. We switch to another model from Huggingface developed by Xu Ming [9]. This model will calculate the similarity score between two sentences. We choose 80% as the minimum score to be considered to share similar body parts and 20% as the maximum score for a pair to be considered not sharing similar body parts. We set this threshold by manually checking the outcome pairs. For a score of more than 90%, two sentences in a pair are the same, which we are not looking for it.



出血 痔疮 肛裂 便血 患者 治	痔疮 治疗 消化 肛管 痔疮栓	1

Figure 1. Here is one example of what we consider as sharing the same body part.



伤口 医院 注射 破伤风抗毒素	月经 月经 排卵	0

Figure 2. Here is one example of what we consider as not sharing the same body part.

Figure 1 corresponding meaning in English	Figure 2 corresponding meaning in English
Bleeding, hemorrhoids, anal fissure, blood in the stool, patients	wound, hospital, injection, tetanus antitoxin
hemorrhoids, treatment, digestion, anal canal, hemorrhoid suppository, hemorrhoid cream	menstruation, menstruation, ovulation

Since the data size is too large and it took too long to train, we sample our data randomly to get a timely result. We obtain a dataset containing **24000** pairs of sentences and their label. 20% of the pairs are labeled sharing the same body part.

Computational Requirements

We run it on a regular student laptop (multi-platform or using a VirtualBox virtual machine). The minimum hardware configuration required is a dual-core processor and 2GB of RAM (running memory). For dataset storage, a minimum of 200MB of additional storage space is required for the dataset and training set.

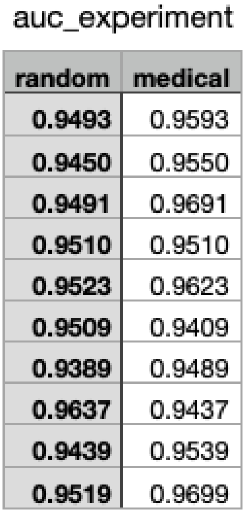
Results

To make sure that the results of our experiments are not outliers, we run each experiment at least ten times and use the mean results as our final answers.

Result 1

The mean AUC score of the CNN model using medical concept vectors is **95.54**% with a standard deviation of 0.0098. The mean AUC score of the CNN model using a randomly initialized vector is **94.96**% with a standard deviation of 0.0065. The mean AUC score of the CNN model using medical concept vectors is **0.58**% higher than the mean AUC score of the CNN model using a randomly initialized vector. The Table 3 showing the AUC score of each round.

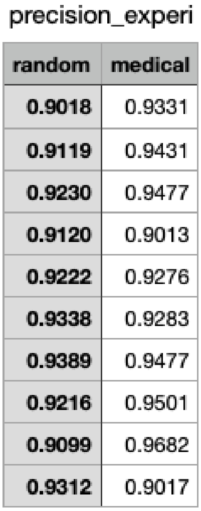
[8] https://github.com/alibaba-research/ChineseBLUE
[9] https://github.com/shibing624/text2vec



auc_experiment

random	medical
0.9493	0.9593
0.9450	0.9550
0.9491	0.9691
0.9510	0.9510
0.9523	0.9623
0.9509	0.9409
0.9389	0.9489
0.9637	0.9437
0.9439	0.9539
0.9519	0.9699

Figure 3. AUC Experiment.



precision_experi

random	medical
0.9018	0.9331
0.9119	0.9431
0.9230	0.9477
0.9120	0.9013
0.9222	0.9276
0.9338	0.9283
0.9389	0.9477
0.9216	0.9501
0.9099	0.9682
0.9312	0.9017

Figure 4. Precision Experiment.

Result 2

The mean precision of the CNN model using medical concept vectors is **93.49**% with a standard deviation of 0.0213. The mean precision of the CNN model using a randomly initialized vector is **92.06**% with a standard deviation of 0.0118. The mean precision of the CNN model using medical concept vectors is **1.43**% higher than the mean precision score of the CNN model using a randomly initialized vector. The Table 4 showing the precision of each round.

Result 3

The mean accuracy of the Doc2Vec model is **82.34**%, which is far lower than the accuracy of Doc2Vec mentioned in the paper. We believe this result is incorrect because the precision of the Doc2Vec model is close to the distribution of our dataset. More specifically, it seems like Doc2Vec is marking all data as not sharing body parts. Furthermore, this will gives Doc2Vec and accuracy close to 80%.

Conclusion and Discussion

Collectively we do not think we achieve 100% complete replication of experiments and conclusions in the original paper. The first reason is that we do not have access to appropriate datasets. We do not have access to the dataset used by the authors, but there are also few Chinese medical datasets. Besides the shortage of data, our lack of medical knowledge also prevents us from labeling data correctly. We have to build a path to circle the barrier of knowledge. We do feel lucky that we still accomplish something in this project.

The data cleaning and processing part are easy as many tools are built for us to use. The part of building the CNN model is also not that hard. The paper gives the parameter for the CNN model. Also, it is a standard CNN model used to detect similarities between sentences. Besides the parameter, the tricks of loading data and batching them increase the training speed.

The difficulty of our experiments comes from two parts. The first one lacks usable data. As we mentioned before, we spent lots of time finding data suitable for our experiment. The second part is that description of the experiments is unclear. In the original paper, those researchers hired physicians to label their data. Only kappa scores between those physicians are provided, which provides no information for us to label our data. Besides labeling data, the paper provides almost no detail regarding baseline algorithms the model is testing against. The paper only mentions the type of algorithms, such as Doc2Vec, LSTM, and LDA. No further information regarding hyperparameters is provided.

Suppose the model needs to be ported to similar Chinese medical terminology for similarity detection. In that case, we recommend that the first step is obtaining enough data with many specialized terms. This will significantly improve the accuracy of the algorithm. Secondly, in constructing the model, we need to pay attention to the use of hyperparameters of the convolutional neural network algorithm. We use similar parameters as the original authors to support the article's theory and conclusions. Overall it is difficult to repeat the implementation of this article, but we can still advance a lot if we combine a lot of other methods.

Data Availability Statement. The data that support the findings of this study are openly available in Github at https://github.com/liutiantian233/CNN4Medical.

References

Fernández-Suárez, X. M., Rigden, D. J., & Galperin, M. Y. (2014). The 2014 Nucleic Acids Research Database Issue and an updated NAR online Molecular Biology Database Collection. *Nucleic acids research*, **42**, D1–D6. https://doi.org/10.1093/nar/gkt1282.

Vishrawas Gopalakrishnan, Kishlay Jha, Wei Jin, Aidong Zhang (2019). A survey on literature based discovery approaches in biomedical domain. *Journal of Biomedical Informatics*, **93**, 103141. https://doi.org/10.1016/j.jbi.2019.103141.

Tao Zheng, Yimei Gao, Fei Wang, Chenhao Fan, Xingzhi Fu, Mei Li, Ya Zhang, Shaodian Zhang & Handong Ma (2019). Detection of medical text semantic similarity based on convolutional neural network. *BMC Medical Informatics and Decision Making*, **19**, 156. https://doi.org/10.1186/s12911-019-0880-2.

Aliaksei Severyn and Alessandro Moschitti (2015). Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, **15**, 373-382. https://doi.org/10.1145/2766462.2767738.

Yin, Wenpeng and Schütze, Hinrich and Xiang, Bing and Zhou, Bowen (2016). ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs. *Transactions of the Association for Computational Linguistics*, **4**, 259-272. https://doi.org/10.1162/tacl_a_00097.

Deerwester, Scott and Dumais, Susan T. and Furnas, George W. and Landauer, Thomas K. and Harshman, Richard (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, **41**, 391-407. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

Blei, David M. and Ng, Andrew Y. and Jordan, Michael I. (2003). Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, **3**, 993–1022. https://dl.acm.org/doi/10.5555/944919.944937.

Quoc Le and Tomas Mikolov (2014). Distributed representations of sentences and documents. *Proceedings of the 31st International Conference on International Conference on Machine Learning*, **32**, II–1188–II–1196. https://dl.acm.org/doi/10.5555/3044805.3045025.

Mueller, Jonas and Thyagarajan, Aditya (2016). Siamese Recurrent Architectures for Learning Sentence Similarity. *Proceedings of the AAAI Conference on Artificial Intelligence*, **30**, 1. https://ojs.aaai.org/index.php/AAAI/article/view/10350.

Yonghui Wu, Min Jiang, Jianbo Lei, Hua Xu (2015). Named Entity Recognition in Chinese Clinical Text Using Deep Neural Network. *Studies in health technology and informatics*, **216**, 624–628. https://doi.org/10.3233/978-1-61499-564-7-624.

Aditya Grover and Jure Leskovec (2016). Node2vec: Scalable Feature Learning for Networks. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, **22**, 855–864. https://doi.org/10.1145/2939672.2939754.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean (2013). Distributed Representations of Words and Phrases and their Compositionality. *27th Annual Conference on Neural Information Processing Systems 2013*, **27**, 3111–3119. https://doi.org/10.48550/arXiv.1310.4546.

Ruixuan Luo, Jingjing Xu, Yi Zhang, Xuancheng Ren, Xu Sun (2019). PKUSEG: A Toolkit for Multi-Domain Chinese Word Segmentation. *ArXiv*, **1906.11455**, 1. https://doi.org/10.48550/arXiv.1906.11455.

Ningyu Zhang, Qianghuai Jia, Kangping Yin, Liang Dong, Feng Gao, Nengwei Hua (2020). Conceptualized Representation Learning for Chinese Biomedical Text Mining. *The 13th ACM International WSDM Conference*, **2008.10813**, 1. https://doi.org/10.48550/arXiv.2008.10813.