Clinical Text Classification with Word Embedding Features vs. Bag-of-Words Features

Yijun Shao
Biomedical Informatics Center
George Washington University
Washington, USA
yshao@gwu.edu

Stephanie Taylor
Health Services Research and Development
VA Greater Los Angeles Healthcare System
Los Angeles, USA
stephanie.taylor8@va.gov

Nell Marshall
Health Services Research and Development
VA Palo Alto Health Care System
Palo Alto, USA
nell.marshall@va.gov

Craig Morioka
Department of Radiology
VA Greater Los Angeles Healthcare System
Los Angeles, USA
craig.morioka@va.gov

Qing Zeng-Treitler
Biomedical Informatics Center
George Washington University
Washington, USA
zengq@email.gwu.edu

***Abstract*—Word embedding motivated by deep learning have shown promising results over traditional bag-of-words features for natural language processing. When trained on large text corpora, word embedding methods such as word2vec and doc2vec methods have the advantage of learning from unlabeled data and reduce the dimension of the feature space. In this study, we experimented with word2vec and doc2vec features for a set of clinical text classification tasks and compared the results with using the traditional bag-of-words (BOW) features. The study showed that the word2vec features performed better than the BOW-1-gram features. However, when 2-grams were added to BOW, comparison results were mixed.**

Keywords—word embedding; natural language processing; clinical text mining;

I. INTRODUCTION

Recently, a class of special machine learning methods called "deep learning" has gained much attention because it has shown superb performance over traditional machine learning methods such as support vector machine, random forest, etc. on tasks like image classification, speech recognition, [1-3] etc. One reason is that deep learning not only learns to discriminate instances from different classes just as the traditional machine learning methods do but also learns to construct useful higher level features from the raw data which for the traditional methods is usually done manually [4]. Because the higher level features are learned from data, the machine has a chance to construct better features than human and consequently outperforms the traditional learning methods.

A deep-learning motivated feature representation in natural language processing (NLP) is the word embedding, or word2vec [5]. Traditionally, words are represented as one-hot vectors, so all words are equally distant with each other and this representation carries no semantic or syntactic meanings. Word2vec represents words as real-valued vectors in a vector space of relatively low dimensions (compared to the vocabulary size). The advantage of word2vec representation over the traditional representation such as one-hot vectors is that the vectors carry semantic or syntactic information. In particular, words that are close in the vector space are also semantically or syntactically close, and vector subtractions and additions represent semantic subtractions and additions (e.g., "king" – "man" + "woman" = "queen"). Although word2vec is one of the dimension reduction methods including Principle Component Analysis (PCA) [6], Latent Semantic Analysis (LSA) [7] and Latent Dirichlet Allocation (LDA) [8], the vector subtractions and additions properties of word2vec has made it more attractive to NLP researchers.

An extension of word2vec is doc2vec [9], which was motivated by the need to represent documents (or other kinds of pieces of texts such as paragraphs or sentences) as real-valued vectors in a similar way as word2vec represents words as vectors. For many NLP tasks such as text classification and document retrieval, the target units of interests are not individual words but documents. The common approaches include representing a document as a sum or average of the vectors of all the words in it, but these seem more ad-hoc than natural. By faking the document as a "word" within the context of all words in the document, the word2vec algorithm can be used to obtain vector representations for both the words and the document. This provides a way of document representation that is more natural than a simple sum or average.

Machine learning is widely used in clinical NLP such as text classification, information extraction, [10-14] etc. There is significant interest in the clinical NLP community to apply deep learning and improve text classification performance [15, 16].

When performing machine learning for NLP, the common first step is to extract features such as bag of words or bag of n-grams. For convenience, we will refer to both bag of words and bag of n-grams as bag of words (BOW). Then these features were represented as one-hot vectors so that machine learning methods can process the features for training and classification. Given the success of word2vec features in tasks such as sentiment analysis [17], a research question is if they will

978-1-5386-5035-6/18/$31.00 ©2018 IEEE

perform better than BOW features. On the other hand, some text classification tasks reported in literature are quite "coarse" (e.g. sentiment analysis), while clinical text classifications can be fine grained (e.g. identifying the usage of specific therapies). In the fine-grained classification, there may be a need to differentiate "is" from "was" or "recommended" from "prescribed," while word2vec are likely to represent them using similar vectors.

To explore the use of word2vec/doc2vec features in clinical text classifications, we compared them with the traditional bag-of-words (BOW) features in a series of experiments. The text classification performance is reported and the potential reasons for the performance difference is discussed.

II. Methods

Our experiments performed classifications using labeled snippets from each individual modality as well as using those from all modalities combined. In both cases, the goal is to differentiate current user from all other cases (i.e. past user, nonuser, or uncertain).

A. Dataset

The dataset for this study was generated from the Veterans Affairs (VA) electronic medical records (EMR) stored in the Veterans Administration Informatics and Computing Infrastructure (VINCI) database. The dataset was developed as part of a project which studies the use of Complementary and Alternative Medicine (CAM) among the veterans. To extract the CAM utilization documented in clinical notes, we annotated a dataset using a random sample of clinical notes from VINCI. Each note in the data set contained at least 1 pre-defined CAM-related keyword. For example, for Acupuncture, the keywords were "acupuncture", "ACUP", "needling". When a keyword was found in a note, a snippet composed of the keyword together with 30 words before and 30 words after was extracted.

The CAM snippet dataset was further divided into 6 modalities: Acupuncture, Biofeedback, Guided Imagery, Meditation, Tai-Chi and Yoga. For each modality, a small subset (n=500~600) of the snippets was selected for human annotation. The human annotated data were then used to develop a set of NLP extraction tools. An annotation guideline was developed and iteratively revised by the authors of this paper. Each snippet was labelled as "current user," "planned/recommended," "uncertain," "past user," and "nonuser." Given our interest in current CAM users, we grouped the original multiple category annotation labels into binary labels: "current user" (positive) vs. "all other cases" (negative). "Current user" means that the snippet shows a patient was a current CAM user at the time when the note was taken. The annotation was first performed by a dedicated annotator and subsequently reviewed according to the guideline and revised by 2 other team members. Questions and disagreement were resolved through discussion.

B. Features

We first tokenized the snippets by converting all upper cases to lower and removing all punctuations and numbers. Then we used the remaining words to build features.

For BOW features, we considered both 1-grams (i.e., words) and 2-grams. The 2-grams were two adjacent words that were originally (i.e., before removing punctuations and numbers) separated by only white spaces. Because of the large number features of this type, we conducted feature selection. Features with high discriminative power were selected. The discriminative power of a feature w with respect to a category c was defined [18] to be

$$D_c(w) = \frac{1 - p(c)}{1 - p_w(c)}$$

where $p(c)$ is the proportion of snippets in category c among the training snippets, and $p_w(c)$ is the proportion of snippets in category c among the training snippets containing the feature w. We selected features which occurred in >=2 snippets and had discriminative power >= 1.5 in either positive or negative category.

For word2vec/doc2vec features, we used an implementation provided by a Python programming package called "gensim" [19]. For classification on an individual modality, word2vec/doc2vec was trained only on the snippets from that modality, and for the classification on the combination of all modalities, word2vec/doc2vec was trained on the combination of all snippets. In each of the cases, we trained word2vec/doc2vec on both the labeled and the unlabeled snippets, with the number of unlabeled snippets being far bigger than labeled (Table I). We believed that the training on the large number of unlabeled snippets would provide additional knowledge to help the learning on the labeled snippets and improve the classification performance. This step of the knowledge acquisition is absent from the generation of BOW, which only utilized the labeled data. For word2vec/doc2vec, the dimension of the output vectors must be given beforehand. The number is typically in hundreds, and we chose 300 as the dimension. To generate the vectors, we used the continuous bag-of-words (CBOW) [5] algorithm for word2vec and distributed memory [9] algorithm for doc2vec. The distributed memory algorithm is a straightforward extension of CBOW. The window size for context was set to be 10, i.e., 5 words before and 5 words after the center word formed the context.

C. Text representation and classification

Since the classification was done on the snippets, we needed to represent them as vectors. For BOW, each snippet was represented as a binary vector: each dimension corresponded to a feature (1- or 2-grams) and the value (either 1 or 0) indicated the presence/absence of the feature in the snippet. For word2vec features, we represented each snippet as the average of the word vectors for all the words occurring in the snippet. For doc2vec, since each snippet was given a vector representation by the algorithm automatically, we simply used that vector representation for the snippet.

We used support vector machine (SVM) [20] for classification of the snippets. The vectors representing the snippets were supplied to SVM for learning and classifying. We chose SVM because it has been shown shown to be among the best machine learning algorithms for text classification, and also because our emphasis was on comparing the feature

representations rather than the classification algorithms. The classification was binary: "current user" (positive) vs. "all other cases" (negative). For fair comparison and simplicity, we used linear SVM [21] (i.e., SVM with linear kernel) in all experiments.

D. Experiments

First we conducted a comparison among 4 types of features: 1) BOW-1-gram, 2) BOW-1,2-gram, 3) word2vec, 4) doc2vec. This comparison was done on the combined set of all 3100 snippets from all 6 modalities, therefore, the word2vec and doc2vec features were trained on all the snippets (including both labeled and unlabeled) from all 6 modalities. For word2vec and doc2vec, we set the vector dimension to be 300.

Next we conducted a comparison between 3 types of features: 1) BOW-1,2-gram, 2) word2vec, and 3) doc2vec. We omitted the features of BOW-1-gram from this round because BOW-1-gram features usually perform worse than BOW-1-and-2-gram features. This comparison was done on each of the 6 modalities, and therefore the word2vec and doc2vec features were also trained on the snippets (including both labeled and unlabeled) from each modality.

We used 10-fold cross validation to measure the classification performance. For experiments using BOW features, the extraction and selection of features were done separately for each fold on the training data. Therefore, the set of features (i.e. 1- or 2-grams) can be different across the different folds within a single experiment. In contrast, the word2vec/doc2ec features were fixed once they were generated.

We calculated performance metrics including area under curve (AUC), accuracy, sensitivity and specificity. We chose AUC to be the primary metric and accuracy as secondary metrics. The SVM classifiers were tuned to optimize AUC first. With the best configuration for AUC, the accuracy was calculated for all thresholds on the SVM scores and the best value was chosen for the final classification model. The reported performance were micro-averaged over the 10-folds for each metric.

III. Results

B The dataset used for this study contained snippet texts extracted from clinical notes. The number of annotated and un-annotated snippets are shown in Table I.

The word vectors learned by word2vec carried semantic or syntactic information. Table II shows some examples of neighboring words from the word2vec model trained on the combined set with 600 dimensions. We used cosine similarity for ranking similarity. Most similar words identified by word2vec were close in semantics, though some would not be considered close in the context of certain clinical classification tasks (e.g., "no" and "yes" in the last row of Table II).

We experimented with 4 feature types including BOW-1-gram, BOW-1,2-gram, word2vec and doc2vec for classification on the combined set of snippets from all 6 modalities. In Table III, we reported the AUC and accuracy, sensitivity and specificity.

In the comparison of BOW-1, 2-gram, word2vec and doc2vec features on the 6 individual modalities, word2vec performed better in 5 modalities and BOW performed bettered in one of them. The doc2vec features performed worse in all 6 experiments. The results are shown in Table IV.

We further compared the classification errors of the 3 feature types by calculating the proportions of the snippets that classifications of two feature types were both correct, only one was correct, and neither were correct. The results are shown in Fig. 1. The proportions were calculated based on all the 3100 snippets from all 6 modalities, but the classifications were collected from the models trained on individual modalities.

TABLE I. THE BASIC CHARACTERISTICS OF THE DATASET FOR STUDY

Modalities	Number of Annotated Snippets	Number of Un-Annotated Snippets
Acupuncture	500	375,547
Biofeedback	500	136,011
Guided Imagery	500	62,103
Meditation	600	990,917
Tai-Chi	500	80,996
Yoga	500	200,838
Total	*3100*	*1,846,412*

TABLE II. TABLE TYPE STYLES

Word	Most similar words
acupuncture	accupuncture, acupunture, accupunture, injections, chiropractic
recommend	suggest, recommended, reccomend, consider, agree
interested	intersted, interest, uninterested, amenable, interesed
pain	pains, headache, apin, stress, anxiety
is	was, are, remains, feels, seems
pt	vet, veteran, patient, he, she
scheduled	rescheduled, scheudled, arranged, sched, sheduled
not	never, nto, nothing, it, he
will	would, should, may, could, can
received	recieved, provided, undergone, receieved, receives
have	ve, has, havent, feel, haven
yes	x, no, none, xyes, n

TABLE III. PERFORMANCE OF EXPERIMENTS DONE ON THE COMBINED SET OF ALL MODALITIES

Feature Type	AUC	Accuracy	Sensitivity	Specificity
BOW-1-gram	0.879	0.809	0.846	0.758
BOW-1,2-gram	**0.892**	**0.815**	0.862	0.751
Word2vec	0.887	0.812	0.864	0.741
Doc2vec	0.822	0.767	0.842	0.664

TABLE IV. PERFORMANCE OF EXPERIMENTS DONE ON INDIVIDUAL MODALITIES

Modality	BOW-1,2-gram		Word2vec		Doc2vec	
	AUC	Accuracy	AUC	Accuracy	AUC	Accuracy
Acupuncture	0.918	0.858	**0.921**	**0.864**	0.741	0.692
Biofeedback	0.864	0.804	**0.892**	**0.822**	0.778	0.724
Guided Imagery	**0.910**	**0.854**	0.902	0.834	0.827	0.776
Meditation	0.828	0.833	**0.860**	**0.843**	0.813	0.812
Tai-Chi	0.878	0.818	**0.893**	**0.844**	0.760	0.748
Yoga	0.856	0.798	**0.866**	**0.840**	0.761	0.750

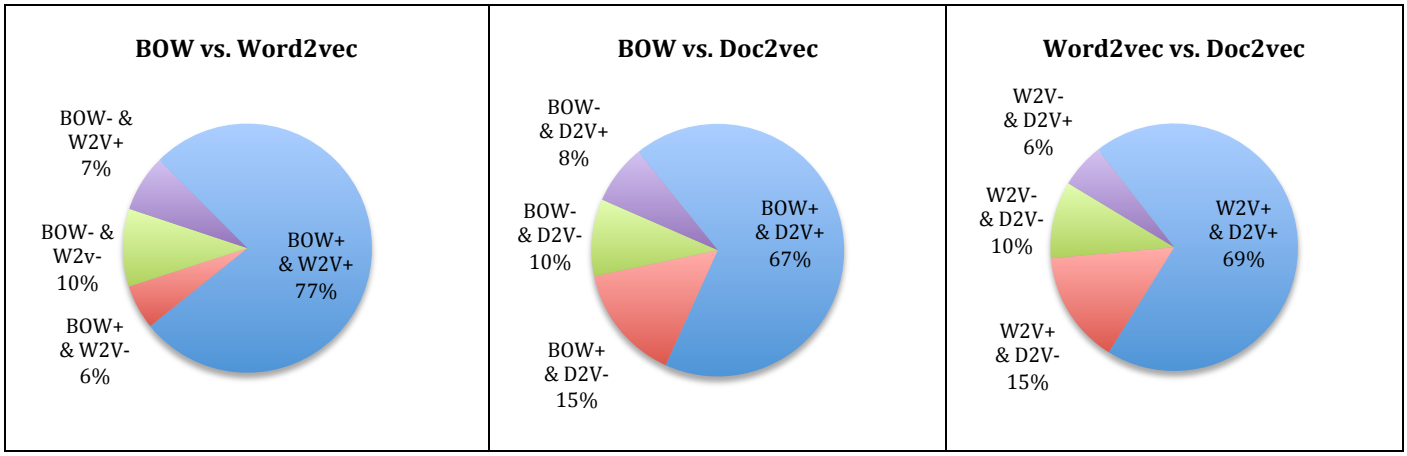

Fig. 1. Comparison of the classification results on all 6 modalities (total of 3100 snippets) using the 3 types of features: BOW-1,2-gram vs. word2vec vs. doc2vec. The "+" signs indicate "correct classifications" and the "-" signs indicate "incorrect classifications" (e.g., "BOW+ & W2V-" means the snippets that BOW classified correctly and word2vec classified incorrectly)

IV. DISCUSSIONS AND CONCLUSIONS

A In this study, we explored the word embedding (word2vec/doc2vec) features in clinical text classification and compared them with the traditional bag-of-words (BOW) features. It was not surprising to see that BOW-1,2-gram features performed better than BOW-1-gram as shown in Table III, since 2-grams captured word order information that is missing in BOW-1-gram, and the word order information is sometimes critical for classification [22].

The results also showed that word2vec features were better than the BOW-1-gram features. Like BOW-1-gram, the word order information was lost when a simple averaging method was used to represent the snippets as vectors based on word2vec. The improvement of word2vec over BOW-1-gram was likely due to the external knowledge acquired by word2vec from training on the large number of unlabeled snippets. Such knowledge can be particularly important when the labeled data set was small (for each individual module, we had only 500-600 labeled snippets) and the unlabeled data size was much larger (Table I). In Table II, one can see that word2vec successfully captured some synonyms and misspellings as close neighbors, which likely benefited the classification performance. In contrast, BOW treated these synonyms and misspelling as totally different words.

We also observed that word2vec regarded antonyms as similar words as well, e.g. "interested" and "uninterested", "have" and "havent", and "yes" and "no". These words were represented as similar vectors possibly because their context words (from a small context window) were often similar, and the learning of word2vec was completely based on contexts. Such knowledge might undermine the performance of classification, because "yes" might indicate a current CAM user while "no" might indicate the opposite. For coarse classifications such as sentiment analysis, antonyms do not pose as a serious problem because for example "king" and "queen" can be considered similar when identifying news regarding the royal family. For clinical text classification, further studies are needed for us to distinguish antonyms from synonyms.

The doc2vec features had the worst performance in all experiments. It was not too surprising to us as similar phenomenon has been observed by other researchers [23, 24]. However, it was not clear why it was so, because the process of generating vectors representing documents was less straightforward and it difficult to find what information was missing compared to BOW-1-gram in the process. In Fig. 1, we compared the classification results between BOW-1,2-gram, word2vec and doc2vec. One can see that even though doc2vec performed the worst, it still made correct

classifications on some snippets on which BOW-1,2-gram and wor2vec did not.

Comparing BOW-1,2-gram to word2vec, we do not see an obvious winner. On the larger set of combination of all 6 modalities, BOW-1,2-gram had better performance (Table III), while on the smaller sets of individual modalities, word2vec performed better in 5 our of 6 cases (Table IV). The only modality (Guided Imagery) on which word2vec was worse was the one that had the least number of unlabeled snippets. Since word2vec relied much on acquiring external knowledge from unlabeled data, this seems to suggest that having training word2vec on larger unlabeled data is more beneficial. On the other hand, the results of Table III may hint that, when the size labeled data becomes larger, the benefit of training word2vec on larger unlabeled dataset is diminished.

There are some limitations in this study. First, when using word2vec features, we represented each snippet as an average over all the vectors for words occurring in the snippet. We could have used some weighting schemes such as term-frequency-inverse-document-frequency (TF-IDF) or IDF alone, but whether a weighting scheme is beneficial remains to be studied. Second, word2vec model has an important parameter – the dimension of the vector – which must be set before training. This parameter may influence the performance, but there is no rule of thumb for choosing the parameter. People usually experiment with a number of choices of the dimension and pick the one giving the best result. In our study, we used 300 as the dimension for word2vec, because our experimentation with other dimensions did not reveal significant difference. This parameter, however, can be further refined in future studies.

For future work, we would like to experiment with representing a snippet as a sequence of word2vec vectors, so that the full word-order information is preserved. Then we can use recurrent neural network for the classification, or even convolutional network if we interpret the sequence of vectors as a 2-D image [24, 25]. It is also an interesting direction to investigate how we can combine the strengths of BOW-1,2-gram, word2vec and doc2vec features to make the performance better than any of them alone.

REFERENCES

[1] LeCun Y, Bengio Y and Hinton G. Deep learning. Nature. 2015; 521: 436-44.

[2] Krizhevsky A, Sutskever I and Hinton G. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems. 2012, p. 1090-8.

[3] Mikolov T, Deoras A, Povey D, Burget L and Cernocky J. Strategies for training large scale neural network language models. Automatic Speech Recognition and Understanding. 2011, p. 196-201.

[4] Zeiler MD and Fergus R. Visualizing and understanding convolutional networks. Computer Vision - ECCV. 2014.

[5] Mikolov T, Sutskever I, Chen K and Corrado G. Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems. 2013.

[6] Jolliffe IT. Principle Component Analysis. Springer-Verlag, 1986.

[7] Dumais ST. Latent semantic analysis. Annu Rev Inform Sci. 2004; 38: 189-230.

[8] Blei DM, Ng AY and Jordan MI. Latent Dirichlet allocation. Journal of Machine Learning Research. 2003: 993-1022.

[9] Le Q and Mikolov T. Distributed representations of sentences and documents. The 31st International Conference on Machine Learning. Beijing, China: JMLR: W&CP, 2014.

[10] Sasaki Y, Rea B and Ananiadou S. Multi-topic aspects in clinical text classification. IEEE International Conference on Bioinformatics and Biomedicine. 2007.

[11] Garla VN and Brandt C. Ontology-guided feature engineering for clinical text classification. Journal of biomedical informatics. 2012; 45: 992-8.

[12] Garla V. Kernel methods and semantic techniques for clinical text classification. PhD Thesis. Yale University, 2012.

[13] Schuemie MJ, Sen E, t Jong GW, van Soest EM, Sturkenboom MC and Kors JA. Automating classification of free-text electronic health records for epidemiological studies. Pharmacoepidemiology and drug safety. 2012; 21: 651-8.

[14] Redd D, Kuang J, Mohanty A, Bray BE and Zeng-Treitler Q. Regular expression-based learning for METS value extraction. AMIA Joint Summits on Translational Science Proceedings. 2016 (In press).

[15] Minarro-Giménez JA, Marín-Alonso O and Samwald M. Exploring the application of deep learning techniques on medical text corpora. E-Health – For Continuity of Care: Proceedings of MIE2014. 2014.

[16] Turner CA, Jacobs AD, Marques CK, et al. Word2Vec inversion and traditional text classifiers for phenotyping lupus. BMC medical informatics and decision making. 2017; 17: 126.

[17] Zharmagambetov SA and Pak AA. Sentiment analysis of a document using deep learning approach and decision trees. Twelvth International Conference on Electronics Computer and Computation (ICECCO). 2015.

[18] Walsh J, Shao Y, Leng J, et al. Identifying axial spondyloarthritis in electronic medical records of United States Veterans. Arthritis Care Res (Hoboken). 2016.

[19] Řehůřek R. Software Framework for topic modelling with large corpora. LREC 2010 Workshop on New Challenges for NLP Frameworks ELRA. 2010, p. 45-50.

[20] Cortes C and Vapnik VN. Support-vector networks. Macchine Learning. 1995; 20: 273-97.

[21] Chang CC and Lin CJ. LIBSVM: A library for support vector machines. Acm T Intel Syst Tec. 2011; 2.

[22] Wu Y, Xu J, Jiang M, Zhang Y and Xu H. A Study of neural word embeddings for named entity recognition in clinical text. AMIA Annual Symposium proceedings / AMIA Symposium AMIA Symposium. 2015; 2015: 1326-33.

[23] Goodwin TR and Harabagiu SM. Deep learning from EEG reports for inferring underspecified information. AMIA Joint Summits on Translational Science. 2017; 2017: 112-21.

[24] Hughes M, Li I, Kotoulas S and Suzumura T. Medical text classification using convolutional neural networks. Studies in health technology and informatics. 2017; 235: 246-50.

[25] Kim Y. Convolutional neural networks for sentence classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014, p. 1746-51.