Vocabulary Domain Prediction for Pathological Report Analysis Using ICD-O3

Sunho Choi, Insoo Kim, Yoojoong Kim, Junhee Seok
School of Electrical Engineering
Korea University, Korea
schoi@korea.ac.kr
dlstn0910@korea.ac.kr
sunbisunbi@korea.ac.kr
jseok14@korea.ac.kr

***Abstract*—Pathology is a basic medical field of diagnosing diseases and describing their conditions that are presented in a medical report form. It is important to understand the medical domain from various medical terms in the reports, such as observation parts, disease conditions and names, and measuring units. In this paper, we apply various machine learning algorithms to predict the domain of untrained terms used in real pathological reports. Here, we focus on the oncology section with ICD-O3. The analysis result shows the possibility and potential usefulness of the domain prediction using medical terms.**

Keywords—Pathology, Text Classification, ICD-O3, Machine Learning, Natural Language Processing

I. INTRODUCTION

A person may have visited a hospital after being injured or infected by a disease more than once in their life. Whatever the reason they visit the hospital, they will receive a documented report that includes the doctor's opinion of their status. This report is a kind of pathological report. Pathology is the study of the causes and effects of disease or injury. Pathology is a basic medical field. In general, all studies done on diseases are called pathology, and pathological report mainly describes the symptoms and conditions of patient in the form of text writings.

Since pathological analysis is used in all medical fields, there can be various expressions about the same word in different reports. For example, some reports may only be written as 'gastritis', while others may note 'inflammation of certain areas of the stomach' or 'chronic inflammation of stomach'. Similarly, there are also words that are not used in other fields but are frequently used in certain medical fields.

Data in text format must be mapped to an ontology code when analyzing with the computer. In order to conduct ontology across the whole field of pathology, understanding of each medical field and responding with the same ontology code when referring to the same thing in a different expression is necessary. However, it is hard to mobilize people because there are only a few people deal with the entire medical field. It is almost impossible for a person to assign all words used in real reports to ontology data. So rather than employing a lot of experts and taking a lot of time as long to make a new dictionary, we need to develop a way to match input words with words in ontology, and a way to add new terms to ontology. In this paper, we first determine whether the input word is classified as a term of the appropriate domain when compared with the corresponding ontology, before developing previously described.

In this paper, 64,140 terms of various diagnostic words and sentences used in oncology called International Classification of Diseases for Oncology (ICD-O3) [1] and words such as measuring unit and places that can be noted in reports are used. These terms were provided in OHDSI Athena standardized vocabularies (http://athena.ohdsi.org). There are 12 domains in the ontology as follows; Condition, Metadata, Observation, Unit, Visit, Type Concept, Payer, Cost, Plan Stop Reason, Plan, Episode, and Sponsor. In the 'Condition' domain, actual tumor data is stored such as 'Adenocarcinoma in situ of skin' (ICD-O3 8140/2-C44.9). The 'Metadata' domain includes words that can be used in the Atlas (https://github.com/OHDSI/Atlas) as a platform for collaborating with Korea University Medical School[1] to build ontology. Detailed examples of other domains and ontology can be found in Athena.

By learning these terms, we have compared algorithms that predict which domain an input word is in. We tested whether algorithms could predict the domains by entering the same ontology data which was not trained (split as a test set) and the actual data that would cover other medical fields than oncology alone.

In this paper, we used algorithms such as SimpleRNN [2] (i.e. Elman Architecture), LSTM [5], and Convolutional LSTM [6, 7], which are forms of Recursive Neural Network [2-4], and also Multinomial naive Bayes method [8].

II. EXPERIMENT METHODS

A. Natural Language Processing

Natural language is the language people use in everyday life. Natural language processing is the process of analyzing the meaning of these natural languages and making them available for processing on computers by statistically, not by rule-based [9]. For natural language processing, the text data must be mapped to certain labels.

This work was supported by the grants of the NRF of Korea (NRF-2017R1C1B2002850) and KHIDI[1](HI19C0360, HI19C0201). The correspondence should be addressed to jseok14@korea.ac.kr

978-1-7281-1340-1/19/$31.00 ©2019 IEEE

B. Recursive Neural Network

Elman architecture [2] considers input data now and input data in the past. This architecture store memory in hidden layers as **i**f they are storing memories in their heads. It has a hidden state $\boldsymbol{h}_t$ which depends on $\boldsymbol{h}_{t-1}$; i.e.

$$\boldsymbol{h}_t = \tanh(\mathbf{W}_{hh}\boldsymbol{h}_{t-1} + \mathbf{W}_{xh}\boldsymbol{x}_t) \qquad (1)$$

where $\boldsymbol{x}_t$ is the input. $\mathbf{W}_{hh}$ and $\mathbf{W}_{xh}$ are weight matrices.

The RNNs output value at time t-1 also affects the RNNs output value at time t. The structure in which the past output is input again is called the feedback structure.

C. Long Short-Term Memory

LSTM is a special kind of RNN that has the ability to perform learning that requires a long dependency period [5]. The LSTM unit consists of a cell with several gates attached. It has the function of saving the information of this cell, retrieving the information of the cell, and maintaining the information of the cell. LSTM is written below as

$$\begin{pmatrix} \boldsymbol{i} \\ \boldsymbol{f} \\ \boldsymbol{o} \\ \boldsymbol{g} \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \mathbf{W} \begin{pmatrix} \boldsymbol{h}_{t-1} \\ \boldsymbol{x}_t \end{pmatrix} \qquad (2)$$

$$\boldsymbol{c}_t = \boldsymbol{f} \odot \boldsymbol{c}_{t-1} + \boldsymbol{i} \odot \boldsymbol{g} \qquad (3)$$

$$\boldsymbol{h}_t = \boldsymbol{o} \odot \tanh(\boldsymbol{c}_t) \qquad (4)$$

where $\boldsymbol{i}$, $\boldsymbol{f}$, and $\boldsymbol{o}$ are input gate, forget gate, and output gate. $\boldsymbol{h}_t$ is output from the LSTM. $\odot$ denotes an element-wise product. $\sigma(\cdot)$ and $\mathbf{W}$ are logistic function and weight matrix.

D. Convolutional LSTM Network

It is already well known that CNN performs as well as RNN in text classification [10]. However, this algorithm does not apply LSTM after CNN, but it replaces all parts of the LSTM expressions that were simple multiplication operations with convolution operations [7]. This means that the number of weights presents in all W in each cell can be significantly less than the LSTM. This is the same effect as when the Fully-Connected Layer is replaced by a Convolutional Layer, which can greatly reduce the weight of the model as a whole.

E. Multinomial Naive Bayes (NB) Method

The Naive Bayes Classification is a kind of probability classifier applying the Bayesian theorem that assumes independence between properties. All classifiers commonly assume that all property values are independent of each other. Naive Bayes is a conditional probability model; i.e.

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})} \qquad (5)$$

where k is the number of classes. $\mathbf{x}$ is the input.

III. Experiment Details

The experiment is designed in four stages. SimpleRNN, LSTM, LSTM with Convolution layer and Multinomial Naive Bayes Method were used for all experiments.

As described in the Introduction section, we used the standardized vocabularies provided from OHDSI Athena. It is the first experiment (Exper 1) to measure the domain prediction accuracy of a test set that was not trained by setting a random test set of 10% separately. Since this experiment trained 90% of the vocabulary, there is a feature that the frequency of the new appearing word will not be high for the remaining 10% test set because the average length of all 64,140 terms is 7.50 and the longest ontology consisted of 27 words, many words are combined rather than a single word.

The vocabulary dictionary extracted from Athena, as shown in Figure 1, has a lot of data on the condition, so the accuracy in the first experiment may not be accurate due to the high ratio to the condition domain in the test set. In the second experiment (Exper 2), the accuracy was measured after removing the values of the true Condition domain from the test set to check the algorithms works well

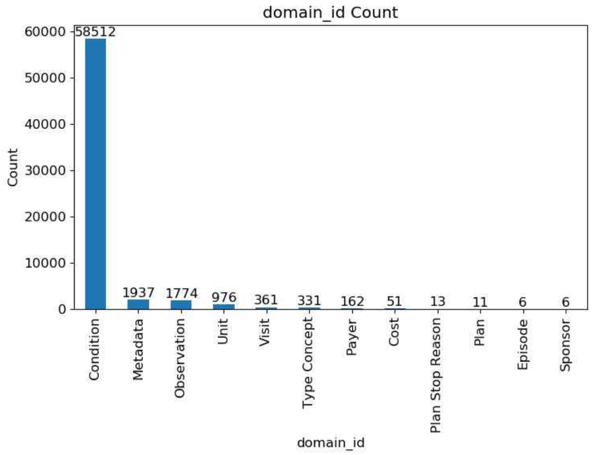

Fig 1. A bar graph measuring the frequency of 64140 standardized vocabulary domains. The condition domain accounts for more than 91% of the total data. On the other hand, the lower 4 domains occupy only 0.05% of the total.

If the previous two experiments were conducted by dividing the train set and test set from the Athena ontology in which similar vocabularies were used, the third and fourth experiments are conducted with terms extracted from the actual Korea University Medical College pathological reports, which has a lot of words that have never appeared in the corresponding ontology.

Only the Condition domain and the Observation domain are labeled as true by a person in the actual pathological documents. So for the third and fourth experiments, only these two domains exist.

The third experiment (Exper 3) used 80 terms from about 1,700 actual pathological documents that were not found in the ontology to see if they were actually good at classifying expressions that were not perfectly matched with ontology.

The fourth experiment (Exper 4), on the other hand, was conducted with all the words extracted from the pathological document, which might overlap with ontology. Thus, the results of this fourth experiment will be closest to accuracy when applied against the actual pathological documentation.

IV. Experiment Results

For the experiments, RNN was designed by Keras [11] and Naïve Bayes by Scikit-Learn [12]. The word index of the train set through Keras's tokenizer was 2,733, and the word index through the TDF of Scikit-Learn was 3,062. The train set and test set division was done through an internal function of Scikit-Learn, where we adjusted the parameters so that at least one value for the lower 5 domains would be included in the train set.

We adjusted the epoch for the RNN experiments to confirm that the overfitting point was the epoch 10 and to match this value in all of the experiments. The following procedure is described in Figure 2.

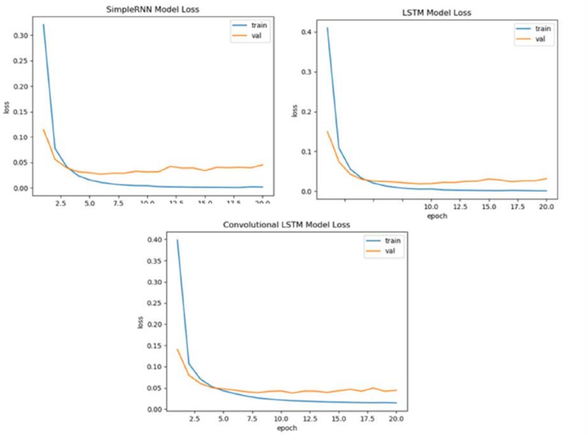

Fig 2 . A train loss and a validation loss graph according to the epochs. The validation rate is 10% for all algorithms.

The number of nodes in the hidden layer was 32 in all, larger than the longest ontology with a length of 27. SimpleRNN and LSTM used one layer each, and the Convolutional LSTM used one 1-D convolution layer and one LSTM layer. Since the output of all algorithms must be one of 12 classes, the output of Dense is 12 and the activation is unified into softmax function.

Table 1 contains the result of all experiments. For the first experiment, Multinomial NB was relatively poor. However, in the second experiment, only the domain with less trained is used as the test set. We can see that Multinomial NB, whose order of words or frequency of training is not significantly more important than other algorithms, came out remarkably well. For the third and fourth experiments with untrained words, the performance of the Multinomial NB is better than other algorithms also.

TABLE I. The Accuracy Result of Experiments

Method	**Experiment**			
	Exper. 1	*Exper. 2*	*Exper. 3*	*Exper. 4*
SimpleRNN	0.9905	0.8894	0.2625	0.3675
LSTM	**0.9925**	0.9124	0.3125	0.4017
ConvLSTM	0.9852	0.8284	0.3500	0.4469
MultiNB	0.9670	**0.9997**	**0.6750**	**0.5092**

V. Conclusion

The association of words within a term was also important, but first, it was an experiment in which we learned that whether the word used within the term existed on the ontology is more important. This can be seen from the fact that LSTM and Convolutional LSTM, normally known to work well in long text analysis, performed worse than NB. Therefore, further research should be conducted to add new terms to the ontology and to properly map terms that have the same meaning but have different expressions.

In addition, although shallow structures are now used, the structures could be improved with the layer being deeper and the number of nodes increasing which could be improved to fit the pathological ontology deployment.

References

[1] A. Fritz, C. Percy, A. Jack, et al. International Classification of Diseases for Oncology (ICD-O), 3rd edition, World Health Organization, Geneva , 2002.

[2] J. Elman, "Finding structure in time," Cognitive science, vol. 14, no. 2, pp. 179–211, 1990.

[3] M. Jordan, "Serial order: a parallel distributed processing approach," Tech. Rep., Univ. of California San Diego, 1997.

[4] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur, "Recurrent neural network based language model," INTERSPEECH, pp. 1045–1048, 2010.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, pp. 1735–1780, 1997

[6] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and Trevor Darrell; "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description," CVPR, pp. 2625-2634, 2015.

[7] S. Xingjian, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. Woo. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," Neural Information Processing Systems, 2015.

[8] A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," AAAI-98 Workshop on Learning for Text Categorization, pp. 41-48, 1998.

[9] Y. Goldberg, "A Primer on Neural Network Models for Natural Language Processing," Journal of Artificial Intelligence Research, vol. 57, pp.345–420, 2016.

[10] X. Zhang, J. Zhao, and Y. LeCun, "Large-Scale Text Classification Methodology with Convolutional Neural Network," Neural Information Processing Systems, 2015.

[11] F. Chollet, Keras. https://github.com/ fchollet/keras, 2015.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol.12, pp.2825–2830, 2011.