Enterprise Master Patient Index Entity Recognition by Long Short-Term Memory Network in Electronic Health Systems

Zhaohui Liang
York University
4700 Keele St., Toronto, Canada
stan79@yorku.ca

Jimmy Huang
York University
4700 Keels St., Toronto, Canada
jhuang@yorku.ca

Jun Liu
Guangzhou Univ Chinese Med
111 Dade Rd, Guangzhou, China
liujun.tcm@163.com

Stephen Chan
Dapasoft INC
111 Gordon Baker Rd, Toronto, Canada
schan@dapasoft.com

Named-entity recognition (NER) is the application of information extraction by artificial intelligence (AI) to locate and classify conceptual entities from natural language into pre-defined categories. In this study, we apply the Long Short-Term Memory network (LSTM) networks to identify the patient entities from the Enterprise Master Patient Index (EMPI). A sample dataset with 300,000 deidentified patient records is used to test the LSTM performance for EMPI entity recognition. The data entries are firstly converted into strings and represented by a Word2Vec model with 200 dimensions. Two LSTM models are developed for the NER recognition problem. The first LSTM model uses a multi-classifier with a softmax function, the second LSTM model uses a two-step classification procedure by binary logistic function. To evaluate the LSTM performance, we use a conventional deep neural network model for comparison, where the Levenshtein distance is used to represent the training data patterns. The classification performance is evaluated by ten-fold cross-validation. The two-step LSTM model has the classification accuracy of 99.82%, which is superior to both the multi-classification LSTM classifier at 61.08% and to the conventional deep neural network at 95.08%. Therefore, we conclude that the new two-step LSTM model provides an accurate and reliable solution to recognize the EMPI patient entities when it is properly configured and trained.

Entity recognition; Long short-term memory (LSTM); Deep learning; Machine learning

1. INTRODUCTION

EMPI is the acronym of Enterprise Master Patient Index. It is also known as Master Patient Index or MPI. MPI matching plays an important role in the multi-database and cross-system integration in eletronic health systems. A recent study finds that the most frequent mismatches include missing data and misspelling in record information. And the deaths due to medical errors are the third leading cause of death in the USA, killing 1,000 people per day [1]. A study in South Korea states that a high-quality EMPI database system can improve the performance of a health information exchange (HIE) for both general purposes and specific purposes [2]. An ideal solution to overcome this difficulty is to apply named entity recognition (NER) with natural language processing by machine learning. The semantic patterns related to the patients in the EMPI system can be learned and presented by appropriate machine learning models. Therefore, a properly trained machine learning model can classify the patient entities during matching in multi-source electronic health data (EHR) integration.

Many complex EHR system applies an EMPI entity database as the patient index to maintain the data consistency and accuracy. The identity information of patients serves as the IDs in large EHR data repositories to query the patients' information cross databases and systems across the EHRs in various hospitals and healthcare institutes. Therefore, it plays a rule as the coordinator to connect information belongs to a specific patient entity or a group of patient entities in complex health information systems. However, an identical patient entity can be represented by different methods. For example, the patient name "John Smith" and "Smith John" can be the identical person. Or how can the system evaluate the records with the same but with some missing identifiers such as SINs are referring to the same entity? Therefore, the EMPI with the NER recognition function contributes and maintains the data consistency in complex EHR systems.

The EMPI system should detect input errors and minor data inconsistency to reduce data redundancy so that the EHR systems can accurately integrate the health information of identical patients from sources.

2. RELATED WORK

The named entity recognition (NER) method has been successfully applied to recognize the entities on Twitter supported by the standard natural language processing (NLP) pipeline [3]. This study reports a 52% improvement in the F1 score when the context is properly represented. Note that this study in 2012 does not apply the current NLP methods such as deep learning with word embedding context representation, which makes the model difficult to accommodate complex semantic representation and less tolerant to noise.

The current NLP methods are mainly based on word representation models, where we can use the semi-supervised learning to reduce data sparsity in the labeled data to improve model generalization. A word representation model uses a vector to map the word features. Bengio et al.find that the NER algorithm had the best great improvement for text classification on F1 score when various word representations are combined [4].

In a complex EHR system with EMPI, we need to perform entity recognition for patient entity matching. NER is crucial for the information retrieval task that seeks to locate and classify named entities in text data into pre-defined categories [5]. In this study, we implement two word representation models to capture both the syntactic and semantic entity patterns from the EMPI connected to multiple EHR database. In the first experiment, we trained an edit-distance pattern matrix to learn the patterns of the correct and empirical typos of the queries by measuring the Levenshtein distance between the correct entries and the typos. Then we use semi-supervised learning to train a conventional deep neural network to perform NER and classification. The second approach is to train a Word2vec embedding for a long-short-term memory neural network to perform NER and classification. By comparing the performance by different classifiers, we are hopeful to discover the best solution for the entity recognition for the EMPI systems.

3. RATIONALE AND APPROACH

3.1 Semantic Representation by Word2vec

To accurate recognize or classify a patient data entity from the EMPI database, we need to recognize the random mismatches and errors created by the users. One method is to present the duplication and error by a matrix containing the edit-distance patterns from empirical data. Then we can train an autoencoder in the semi-supervised learning manner to capture the edit distance patterns represented by the Levenshtein distance (LD) [6]. However, the LD patterns cannot capture semantic patterns that require the information between words.

To capture and learn the semantic patterns, we apply the Word2vec model based on the theory of the language model. Word2vec is a shallow embedding network that use the probabilities to represent the relation of words conditioned on a window of n previous words. It is based on the Markov assumption, where the presence of a word can be computed by the posterior probability of words around it given a specific length of window. The probabilities regarding the presence of certain words can be computed for unigrams and bigrams. Given the centre word c, the surrounding words in a fixed width window are represented by:

$$\boldsymbol{P(o|c)} = \frac{\exp(\boldsymbol{u_o^T} \cdot \boldsymbol{v_c})}{\sum_{w=1}^{v} \exp(\boldsymbol{u_w^T} \cdot \boldsymbol{v_c})} \quad (1)$$

where *o* is the output word index (i.e. the words surrounding the centre word), *c* is the centre word, v_c is the centre vector, and u_o is the output vector. The computed result of *P(o|c)* can be transferred by the Softmax function to compute the posterior probability of word *o* given word *c*:

$$\boldsymbol{P_i} = \frac{\boldsymbol{e^{u_i}}}{\sum_j \boldsymbol{e^{u_i}}} \quad (2)$$

Note that we use the exponent function to yield a positive value and use $\sum_j \boldsymbol{e^{u_i}}$ to normalize the given probability in the vocabulary. Our goal is to maximize the probabilities, which is equivalent to minimize its corresponding negative form to satisfy the convexity of running the stochastic gradient descend (SGD) algorithm:

$$\mathcal{L}(\mathrm{W}) = -\frac{1}{T}\sum_{t=1}^{T} \sum_{\substack{-m \le j \le m \\ j \neq 0}} P\left(w_{t+j} | w_t\right) \quad (3)$$

After the Word2vec is trained in the unsupervised learning manner, we can pipeline the pattern vectors to an LSTM network for entity recognition.

3.2 Recurrent Neural Networks and LSTM

The Recurrent neural network (RNN) is the state-of-the-art deep learning model for language processing. Based on a standard feedforward MLP (multilayer perceptron) network, an RNN is connected by many recurrent loops to add feedback and memory to the networks over time. Thus it can learn and generalize across sequences of inputs rather than individual patterns. The recurrent loops hold the memory and allow an RNN to learn and generalize across sequences of inputs rather than individual patterns.

Starting from the common RNN architecture, the Long Short-Term Memory network (LSTM) adds extra threshold gates to overcome the technical problems of training an RNN, namely vanishing and

exploding of gradients. An LSTM model has many memory blocks that are connected into layers. A block contains gates that manage the block's state and output. A unit operates upon an input sequence, and each gate within a unit uses the sigmoid activation function to control whether they are triggered or not, making the change of state and addition of information owing through the unit conditional. Each unit is like a mini-state machine where the gates of the units have weights that are learned during the training procedure. Compared to a regular RNN, an LSTM has a unique formulation that allows it to avoid the problems of gradient vanishing and gradient exploding that prevent the training and scaling of other RNNs.

Based on the above discussion, we will first train a Word2vec to capture the semantic patterns of the EMPI database. The records in the EMPI tables will be concatenated into plain text and transferred to pattern vectors by the trained Word2vec model. Then the LSTM classifier can be effectively trained by the patterns converted from the Word2vec model and finally acquire reliable classification capacity to recognize the patient entities through the EMPI database.

4. EXPERIMENTS AND RESULTS

4.1 Data Source

An EMPI dataset with 300,000 deidentified patient entity records is acquired from Dapasoft INC., an Ontario government contractor for the maintenance and integration of the Ontario EHR systems. If a new query contains less than or equal to 3 typos or logical errors, it is considered as a minor error, and the system will search for the corresponding row from the database. If a new query has more than three typos or logical errors, it is considered as a major error, and the system should stop searching in the database. The experiment data set has 100,000 rows of correct data, 100,000 rows of data containing minor errors, and 100,000 rows of data containing major errors.

4.2 Experiment Setting

The experiments are implemented in MATLAB R2017b with the official Neural Network Toolbox by MATLAB. The Word2vec embedding model for pattern representation has 200 dimensions. With the Neural Network Toolbox, the LSTM network is composed of a sequence input layer whose input size is equal to the dimension of the word embedding (i.e., 200). To train the LSTM models, we use the stochastic gradient descent (SGD) algorithm with the initial learning rate at 0.01 with a learning rate decay of 0.05 for each epoch. To improve the runtime performance of SGD, we add a momentum of 0.8, and an L2 regularization of 0.001 to overcome overfitting. The training mini-batch is 128.

4.3 Training of Word2vec and LSTM

The data from each row from the EMPI table is converted into a string and concatenated as a single row in the document. The punctuations and stop words are removed before training. The terms are converted to lowercase and tokenized. The patients' records are represented as token vectors. The Word2Vec is set to 200 dimensions and trained for 100 epochs by the stochastic gradient decent (SGD) algorithm.

4.4 Semi-supervised Learning by Autoencoder

We also implement an autoencoder deep neural network to learning the EMPI patient entity patterns represented where the errors are represented by Levenshtein distance (LD). The LD represented pattern matrix is used to train an autoencoder deep neural network with two hidden layers respectively with 30 and ten nodes. The positive saturating linear function is used as the encoder transfer function. The L2 regularization weight is 0.01. A softmax function is added at the end of the neural network architecture.

4.5 Results

In the first experiment, the rows from the EMPI dataset are simply tokenized, concatenated, and converted to characters represented by ASCII codes. The LSTM trained by the character sequence cannot capture the entity patterns. The average classification accuracy is 33.33% in the cross-validation, which implies the LSTM classifier directly trained by character sequence cannot recognize the patient entities from the EMPI data.

In the second experiment, a Word2Vec is trained by the tokenized document records with 100 epochs which generates a vocabulary with the distance matrix of 33,454 terms. Then the 3-class LSTM classifier is fed by the term probabilities pattern sequence computed by the trained Word2vec model from the training documents. The loss value converges well throughout the training process, which implies the LSTM classifier succeeds in minimizing the prediction errors by the training. In the cross-validation for the 3-class LSTM classifier, the average classification accuracy is 79.25%, which implies the LSTM classifier can effectively recognize the EMPI entities, but the performance is not satisfactory. Then we change the strategy to train a two-step LSTM network composed of two binary LSTM classifier. This method simplifies the problem and is hopeful to render better performance. The average accuracy of the two-step LSTM network in the ten-fold cross-validation (30,000 records in the test set) is 99.82%, which implies the two-step LSTM classifier can effectively

recognize the EMPI entities with extremely high reliability. The accuracy and F measure of the 3-class LSTM and the two-step model are shown in ***Table 1***.

Table 1: *Classification Performance of LSTM Networks*

Model	accuracy	F measure	z-test for two proportions	
			Z score	*p*
3-class LSTM	0.7925	0.6108	-82.3027	< 0.01
Two-step LSTM	0.9982	0.9993		

The last step of the experiment is to compare the entity recognition performance by LSTM network with the patterns represented by Word2vec (Word2vec + LSTM), and the deep autoencoder neural network with the patterns measured by Levenshtein distance (LD) distance. The classification performance of the two approaches is shown in ***Table 2***.

Table 2: *Comparison of Two Deep Learning Models*

Model	accuracy	F measure	z-test for two proportions	
			Z score	*p*
LD + autoencoder	0.9626	0.6677	-31.4533	< 0.01
Word2vec + LSTM	0.9982	0.9993		

The result implies that patterns of typos or format errors measured by the LD algorithm are likely to be forgiven by the deep neural networks because the LD algorithm cannot capture the sequence patterns and the relations between words which usually interpreted as semantic meanings. On the other hand, the Word2vec embedding model can capture more details of the text patterns that are later used to train the LSTM networks effectively.

5. CONCLUSION AND DISCUSSION

The LSTM network can properly recognize the patient data entities in the EMPI database when the word embedding method Word2vec properly represents the patterns. If the LSTM classifier determines the new input does not refer to an available EMPI entity, the system stops further processing to save runtime and improve system efficiency. If the new input entity is considered as an available EMPI entity, the system proceeds to the second binary classifier to determine whether the new input has typos or format errors. If it is classified as correct entities, the system performs an equal search by the hashing that maximizes the searching efficiency. If the new query is predicted with minor errors, the system proceeds to similarity search or range searching.

Given that the first binary LSTM classifier has filtered out most of the wrong queries, the overall EMPI system runtime performance is significantly improved with the two-step LSTM entity recognition model. Therefore, the performance of the human-computer interaction and the user experience can be significantly enhanced. Therefore, we conclude that the new two-step LSTM model with Word2vec embedding provides a powerful solution to recognize the EMPI entity similarity when it is properly configured and trained.

6. ACKNOWLEDGMENTS

This research is supported by the National Natural Science Foundation of China (No. 81573827), the Natural Sciences and Engineering Research Council (NSERC) of Canada, and an ORF-RE (Ontario Research Fund - Research Excellence) award in BRAIN Alliance.

7. REFERENCES

[1] Just, BH, Marc D, Munns M, Sandefer R. (2016) Why Patient Matching Is a Challenge: Research on Master Patient Index (MPI) Data Discrepancies in Key Identifying Fields. Perspectives in Health Information Management. 13:1e. eCollection 2016.

[2] Lee M, Heo E, Lim H, Lee JY, Weon S, Chae H, Hwang H, Yoo S. (2015) Developing a common health information exchange platform to implement a nationwide health information network in South Korea. Health Informatics Research. vol.21, issue 1, pp.21-29.

[3] Ritter A, Clark S, Etzioni O. (2012) Named entity recognition in tweets: an experimental study. Conference on Empirical Methods in Natural Language Processing, Jeju Island, Korea, July 12-14, pp. 1524-1534. Association for Computational Linguistics Stroudsburg, PA, USA.

[4] Turian J, Ratinov L, Bengio Y. (2010) Word representations: a simple and general method for semi-supervised learning. The 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden. July 11-16, pp. 384-394. Association for Computational Linguistics Stroudsburg, PA, USA.

[5] Basaldella M, Furrer L, Tasso C, Rinaldi F. (2017) Entity recognition in the biomedical domain using hybrid approach. Journal of Biomedical Semantics. vol. 8, issue 1, pp. 51.

[6] Gilleland M: Levenshtein distance, in three flavors (2009). Merriam Park Software. url: http://www. merriampark. com/ld.htm. (April 15, 2018)