Abstract

Relation Extraction (RE) is one of the most fundamental task of Natural Language Processing (NLP) and Information Extraction (IE). To extract the relationship between two entities in a sentence, two common approaches are (1) using their shortest dependency path (SDP) and (2) using an attention model to capture a context-based representation of the sentence. Each approach suffers from its own disadvantage of either missing or redundant information. In this work, we propose a novel model that combines the advantages of these two approaches. This is based on the basic information in the SDP enhanced with information selected by several attention mechanisms with kernel filters, namely RbSP (Richer-but-Smarter SDP). To exploit the representation behind the RbSP structure effectively, we develop a combined Deep Neural Network (DNN) with a Long Short-Term Memory (LSTM) network on word sequences and a Convolutional Neural Network (CNN) on RbSP.

Experimental results on both general data (SemEval-2010 Task 8) and biomedical data (BioCreative V Track 3 CDR) demonstrate the out-performance of our proposed model over all compared models.

Keywords: Relation Extraction, Shortest Dependency Path, Convolutional Neural Network, Long Short-Term Memory, Attention Mechanism.

Acknowledgements

I would first like to thank my thesis supervisor Prof. Ha Quang Thuy of the Data Science and Knowledge Technology Laboratory at University of Engineering and Technology. He consistently allowed this paper to be my own work, but steered me in the right the direction whenever he thought I needed it.

I also want to acknowledge my co-supervisor Assoc.Prof Chng Eng Siong from Nanyang Technological University, Singapore for offering me the internship opportunities at NTU, Singapore and leading me working on diverse exciting projects.

Furthermore, I am very grateful to my external advisor MSc. Le Hoang Quynh, for insightful comments both in my work and in this thesis, for her support, and for many motivating discussions.

Declaration

I declare that the thesis has been composed by myself and that the work has not be submitted for any other degree or professional qualification. I confirm that the work submitted is my own, except where work which has formed part of jointly-authored publications has been included. My contribution and those of the other authors to this work have been explicitly indicated below. I confirm that appropriate credit has been given within this thesis where reference has been made to the work of others.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone’s copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have fully authorship to improve these materials.

Master student

Can Duy Cat

Table of Contents

Acronyms

Adam Adaptive Moment Estimation

ANN Artificial Neural Network

BiLSTM Bidirectional Long Short-Term Memory

CBOW Continuous Bag-Of-Words

CDR Chemical Disease Relation

CID Chemical-Induced Disease

CNN Convolutional Neural Network

DNN Deep Neural Network

List of Figures

List of Tables

Giới thiệu

Động lực

With the advent of the Internet, we are stepping into a new era, the era of information and technology where the growth and development of each individual, organization, and society is relied on the main strategic resource - information. There exists a large amount of unstructured digital data that are created and maintained within an enterprise or across the Web, including news articles, blogs, papers, research publications, emails, reports, governmental documents, etc. Lot of important information is hidden within these documents that we need to extract to make them more accessible for further processing.

Đặt vấn đề

Relation Extraction task includes of detecting and classifying relationship between entities within a set of artifacts, typically from text or XML documents. Figure 1.1 shows an overview of a typical pipeline for RE system. Here we have to sub-tasks: Named Entity Recognition (NER) task and Relation Classification (RC) task.



A typical pipeline of Relation Extraction system. (Figure 1.1)

A Named Entity (NE) is a specific real-world object that is often represented by a word or phrase. It can be abstract or have a physical existence such as a person, a location, a organization, a product, a brand name, etc. For example, “Hanoi” and “Vietnam” are two named entities, and they are specific mentions in the following sentence: “Hanoi city is the capital of Vietnam”. Named entities can simply be viewed as entity instances (e.g., Hanoi is an instance of a city). A named entity mention in a particular sentence can be using the name itself (Hanoi), nominal (capital of Vietnam), or pronominal (it). Named Entity Recognition is the task of seeking to locate and classify named entity mentions in unstructured text into pre-defined categories.

Difficulties and Challenges (1.3)

Relation Extraction is one of the most challenging problem in Natural Language Processing. There exists plenty of difficulties and challenges, from basic issue of natural language to its various specific issues as below:

•Lexical ambiguity: Due to multi-definitions of a single word, we need to specify some criteria for system to distinguish the proper meaning at the early phase of analyzing. For instance, in “Time flies like an arrow”, the first three word “time”, “flies” and “like” have different roles and meaning, they can all be the main verb, “time” can also be a noun, and “like” could be considered as a preposition.

•Syntactic ambiguity: A popular kind of structural ambiguity is modifier placement. Consider this sentence: “John saw the woman in the park with a telescope”. There are two preposition phases in the example, “in the park” and “with the telescope”. They can modify either “saw” or “woman”. Moreover, they can also modify the first noun “park”. Another difficulty is about negation. Negation is a popular issue in language understanding because it can change the nature of a whole clause or sentence.

Materials and Methods (Chapter 2)

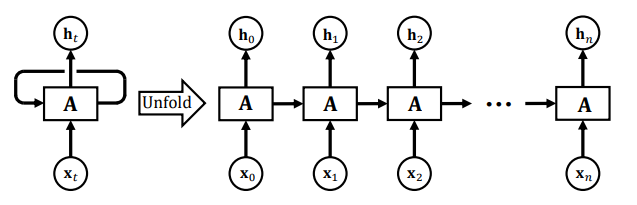
In this chapter, we will discuss on the materials and methods this thesis is focused on. Firstly, Section 3.1 will provide an overall picture of theoretical basis, including distributed representation, convolutional neural network, long short-term memory, and attention mechanism. Secondly, in Section 3.2, we will introduce the overview of our relation classification system. Section 3.3 is about materials and techniques that I proposed to model input sentences to extract relations. The proposed materials include dependency parse tree (or dependency tree) and dependency tree normalization; shortest dependency path (SDP) and dependency unit. I further present a novel representation of a sentence; namely Richer-but-Smarter Shortest Dependency Path (RbSP); that overcome the disadvantages of traditional SDP and take advantages of other useful information on dependency tree.

Theoretical Basis (2.1)

In recent years, deep learning has been extensively studied in natural language processing, a large number of related materials have emerged. In this section, we briefly review some theoretical basis that are used in our model: distributed representation (Subsection 3.1.1), convolutional neural network (Sub-section 3.1.2), long short-term memory (Sub-section 3.1.3), and attention mechanism (Sub-section 3.1.4).

Simple Recurrent Neural Networks (2.1.1)

CNN model are capable of capturing local features on the sequence of input words. However, the long-term dependencies play the vital role in many NLP tasks. The most dominant approach to learn the long-term dependencies is Recurrent Neural Network (RNN). The term “recurrent” applies as each token of the sequence is processed in the same manner and every step depends on the previous calculations and results. This feedback loop distinguishes recurrent networks from feed-forward networks, which ingest their own outputs as their input moment after moment. Recurrent networks are often said to have “memory” since the input sequence has information itself and recurrent networks can use it to perform tasks that feed-forward networks cannot.



Traditional Recurrent Neural Network. (Figure 2.1)

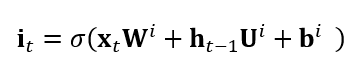
Figure 3.3 illustrates a recurrent neural network and the unfolding across time in its forward computation. This chain-like nature shows that recurrent neural networks closely associated with sequences and lists. The hidden state ht at time step t is calculated based on input at the same time step xt and hidden state of the previous time step ht−1 as follow:



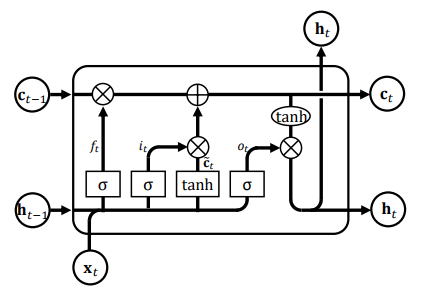
where W, U, and b account for weights and bias that are shared across time; the function f is taken to be a non-linear activation such as ReLU, sigmoid, or tanh. The weight matrices are filters that determine how significant the current input and the past hidden state are. The error that they generate will be returned through back-propagation and used to update their weight during the training phase.

Long Short-Term Memory Unit (2.1.2)

In practice, the vanishing gradient and exploding gradient problem emerged as a major impediment to RNNs performance. Long Short-Term Memory (LSTM) networks are an extension for RNNs, which basically extends their memory. By adding three gates and a memory cell, LSTM (Figure 3.4) calculates the hidden state at time step t as below:

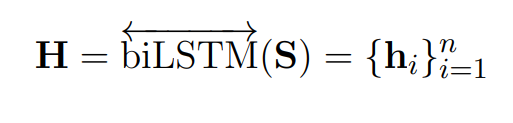


In which, Wi, Ui, Wf, Uf, Wc, Uc, Wo, and Uo are model’s trainable parameters; bi, bf, bc, and bo are bias terms; σ denotes the sigmoid function, and denotes the element-wise product.



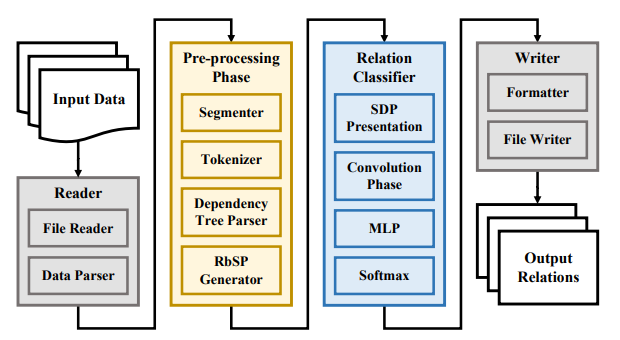
Architecture of a Long Short-Term Memory unit. (Figure 2.2)

To simplify the upcoming explanation, we encapsulate the output of bidirectional Long Short-Term Memory network on a sequence S as follows:



Overview of Proposed System (2.2)

We developed an end-to-end Relation Classification (RC) system that receive the raw input data and export the corresponding relations result. This system is a small framework that contains plug-and-play modules. The overall architecture of RC framework is illustrated in the Figure 3.5.



The overview of end-to-end Relation Classification system. (Figure 2.3)

Proposed system comprises of three main components: IO-Module (Reader and Writer), Pre-processing module, and Relation Classifier. The Reader receives raw input data in many formats (e.g., SemEval 2010 task 8 [29], BioCreative V CDR [65]) and parse them into an unified document format. These document objects are then passed to Pre-processing phase. In this phase, a document is segmented into sentences, and tokenized into tokens (or words). Sentences that contain at least two entities or nominals are processed by dependency parser to generate a dependency tree and a list of corresponding POS tags. A RbSP generator is followed to extract the Shortest Dependency Path and relevant information. In this work, we use spaCy(1 – footnote: spaCy: An industrial-strength NLP system in Python: https://spacy.io) to segment documents, to tokenize sentences and to generate dependency trees. Subsequently, the SDP is classified by a deep neural network to predict a relation label from the pre-defined label set. The architecture of DNN model will be discussed in the following sections. Finally, output relations are converted to standard format and exported to output file.

Experiments and Results (Chapter 3)

Implementation and Configurations (3.1)

Model Implementation (3.1.1)

Our model was implemented using Python version 3.5 and TensorFlow . TensorFlow is a free and open-source platform designed by the Google Brain team for data-flow and differentiable programming across a number of machine learning tasks. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that are used to bring out state-of-the-art in many tasks of ML. TensorFlow can be used in research and industrial environment as well.

Other Python package requirements include:

•numpy

•scipy

•h5py

•Keras

•sklearn

Training and Testing Environment (3.1.2)

Our model’s training and testing experiments are executed on FIT High Performance Computing (FIT-HPC) system. FIT-HPC runs Linux-based operating systems. The configurations of system are listed bellow:

•1 central controller machine

·2 x Intel(R) Xeon(R) CPU E5-2697 v4 @ 2.30GHz

·4 x 16 GB DIMM ECC DDR4 @ 2400MHz

•16 computing machines

•1 machine for database system

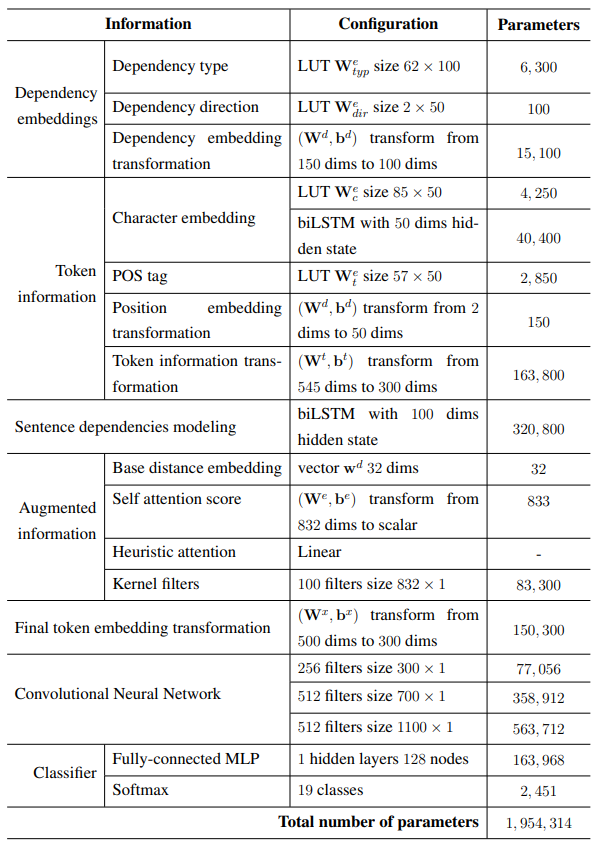
·2 x Intel(R) Xeon(R) CPU E5-2697 v4 @ 2.30GHz

·4 x 16 GB DIMM ECC DDR4 @ 2400MHz

Model Settings (3.1.3)

The tuned hyper-parameters are described in Table 4.1. These hyper-parameters and number of parameters are for reference on SemEval 2010 Task 8 dataset only, the real value may vary depended on testing dataset and some additional techniques.

Configurations and parameters of proposed model. (Table 3.1)

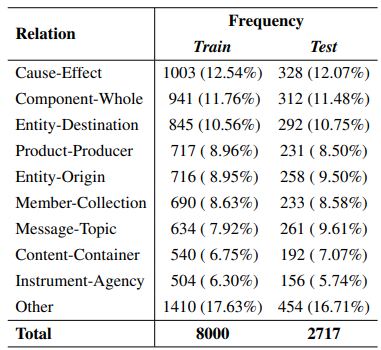


Datasets and Evaluation methods (3.2)

Our model was evaluated on two different datasets: SemEval-2010 Task 8 dataset [29] for general domain relation extraction and BioCreative V Track 3 CDR dataset [65] for chemical-induced disease relation extraction in bio-medical data.

The SemEval-2010 Task 8 contains 10,717 annotated relation classification examples, and is separated into two subsets: 8,000 instances for training and 2,717 for testing. We randomly split 1/10 of the training data for validation dynamically. Table 4.2 shows the distributions of the relation types on this dataset. There are 9 directed relations and one undirected Other class.

Statistics of SemEval-2010 Task 8 dataset. (Table 3.2)



We further investigated this dataset and found that there are 28% of Out-Of-Vocabulary (OOV) words that appear in testing set but do not apprear in training set. More interesting, the percentage of nominal pairs in the testing set that have never appeared in the training set is more than 93%.

Conclusions

In this thesis, we have presented a neural relation extraction architecture with the compositional representation of the SDP. The proposed model is capable of utilizing the dominant linguistic and architectural features, such as word embeddings, character embeddings, position feature, WordNet and Part-Of-Speech tag. In addition, we have presented RbSP, a novel representation of relation between two nominals in a sentence that overcomes the disadvantages of traditional SDP. Our RbSP is created by using multilayer attention to choose relevant information to augment a token in SDP from its child nodes. We also improved the attention mechanisms with kernel filters to capture the features on the context vector.

List of Publications

[1] Duy-Cat Can, Hoang-Quynh Le, Quang-Thuy Ha, and Nigel Collier. “A Richer-but-Smarter Shortest Dependency Path with Attentive Augmentation for Relation Extraction.” In The 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HTL), 2019, (In Press).

[2] Duy-Cat Can, Hoang-Quynh Le, and Quang-Thuy Ha. “Improving Semantic Relation Extraction System with Compositional Dependency Unit on Enriched Shortest Dependency Path.” In The 11th Asian Conference on Intelligent Information and Database Systems (ACIIDS), pp. 140-152, Springer, 2019.

[3] Trang M. Nguyen, Van-Lien Tran, Duy-Cat Can, Quang-Thuy Ha, Ly T. Vu, and Eng-Siong Chng. “QASA: Advanced Document Retriever for Open-Domain Question Answering by Learning to Rank Question-Aware Self-Attentive Document Representations.” In Proceedings of the 3rd International Conference on Machine Learning and Soft Computing, pp. 221-225. ACM, 2019.

References

[1]A. B. Abacha and P. Zweigenbaum, “A hybrid approach for the extraction of semantic relations from medline abstracts,” in International conference on intelligent text processing and computational linguistics. Springer, 2011, pp. 139–150.

[2] E. Agichtein and L. Gravano, “Snowball: Extracting relations from large plaintext collections,” in Proceedings of the fifth ACM conference on Digital libraries. ACM, 2000, pp. 85–94.

[3] A. Airola, S. Pyysalo, J. Bjorne, T. Pahikkala, F. Ginter, and T. Salakoski, “All-¨ paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning,” BMC bioinformatics, vol. 9, no. 11, p. S2, 2008.

[4] I. Augenstein, M. Das, S. Riedel, L. Vikraman, and A. McCallum, “Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications,” in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 546–555.