ENHANCING PERFORMANCE OF THE VIETNAMESE DEPENDENCY PARSING TASK BASED ON ATTENTION MECHANISM AND THE LINGUISTIC PATTERNS

Vinh-Hien Huynh-Dang
21520029@gm.uit.edu.vn

Faculty of Computer Science, University of Information Technology, Vietnam National University - Ho Chi Minh City

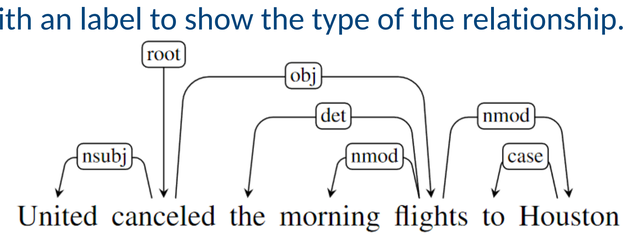
UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Abstract

Our research is about Dependency Parsing, an NLP task on the grammatical structure of language. Dependency Parsing is proven to increase performance of several computing system, including machine translation and question answering. The development of Machine Learning and Neural Networks has made the efficiency of NLP tasks to a new stage, including this tasks. However, the task has a dependence on the language. Most of research from the literature is keen on English or Chinese, and there are only a few of studies based on Vietnamese. On the other hand, similar to other data-related tasks, this task faces the cross-domain problem. In our study, we want to build up a model to solve this task that satisfies at least one in two criteria: having a competitive performance comparing to published research, or having ability to handle cross-domain problem.

Introduction

Dependency Parsing is one of the core tasks in Natural Language Processing (NLP). This task concerns with the grammatical structure of language. Its goal is to map a sentence written in human language into a hierarchical structure called a dependency tree. Each arc in the tree represents a grammatical relationship between a pair of words in the sentence, from a *head* to a *dependent*. Moreover, an arc always comes with an label to show the type of the relationship.



An example of a dependency tree

Task solving methodology

The task solving methodology is usually divided into two main approaches: transition-based and graph-based, whereas the model that convert a sentence into the tree is called a *parser*. A transition-based parser builds the tree by moving words in a sentence sequentially to a stack, and pop up the top two words from it to create an arc whenever possible. Currently, most of research depending on this approach uses a neural network classifier in order to gain score for each transition.
On the other hand, graph-based parser takes the idea from graph theory. First of all, a fully-connected directed graph are built where the nodes are the words in the sentence. Then, the Chu-Liu-Edmonds algorithm is used to get the maximum spanning tree from the graph to become the dependency tree.

Linguistic patterns

The patterns of language in NLP that are commonly used consists of part-of-speech (POS) tagging, name entity recognition (NER), correference resolution, ... which are proven to give informative data for the NLP models to perform better. Consequently, applying them into Vietnamese-related task is considerable.

Cross-domain problem

A dataset used to train Dependency Task is called a *treebank*, where instances of dependency tree are given and the set of arc labels is formalized. As a result, there is a gap between different treebanks, depends on the source of sentences and the formality of labels. For example, in Vietnamese, we have the VnDT dataset and the Vietnamese section from the Universial Dependencies, which have several differences in annotated labels.

Methodology

We propose two main approaches to this task in Vietnamese, including:

Enhancing task performance

For the transition-based parser, we want to implement the solution of [1] where POS tagging is mixed with the representation of words (via an LSTM encoder) to gain score for each transition. As there are extremely rate research on transistion-based parser in Vietnam, we think this is a potential model.
On the other hand, as Vietnamese is a language with flexible word order, graph-based parser can also be taken into account. As the research of Dozat and Manning [2] has great impact on this field, we should take the idea of biaffine attention from them. We suggest replacing the LSTM with next-generation encoder such as BERT or RoBERTa to archive higher performance.
Another point of view is using the unlabeled data to create a semi-supervised model. The solution of [3] is an example where auxiliary models are set to handle raw sentences. Once again, we want to try replacing the legacy encoder.

Handing cross-domain problem

Cross-domain solution is not popular in this field of study. The research of [4] to modify the biaffine attention equation in order to learn the generalization of language is an interesting approach. However, this solution uses large demand of computing performance, which is considerable.

Preliminary Results

We have implemented the solution of [4] and [5] with the VnDT treebank by replacing the LSTM encoder with PhoBERT and recevied the following result.

Model	UAS	LAS
[4]	85.66%	77.88%
[5]	85.60%	77.24%

References

[1] Kiperwasser, Eliyahu, and Yoav Goldberg. 2016. “Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations.” ArXiv.org. July 20, 2016. https://doi.org/10.48550/arXiv.1603.04351.

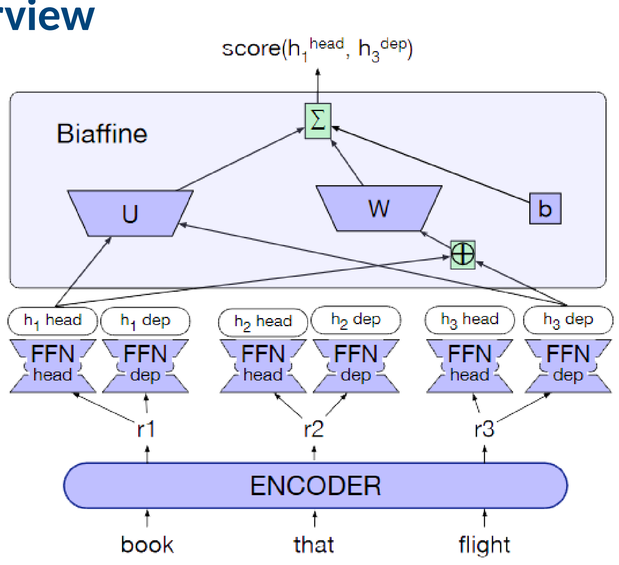
[2] Dozat, Timothy, and Christopher D. Manning. 2017. “Deep Biaffine Attention for Neural Dependency Parsing.” ArXiv.org. March 9, 2017. https://doi.org/10.48550/arXiv.1611.01734.

[3] Clark, Kevin, Minh-Thang Luong, Christopher D. Manning, and Quoc V. Le. 2018. “Semi-Supervised Sequence Modeling with Cross-View Training.” ArXiv.org. September 21, 2018. https://doi.org/10.48550/arXiv.1809.08370.

[4] Tian, Yuanhe, Yan Song, and Fei Xia. 2022. “Enhancing Structure-Aware Encoder with Extremely Limited Data for Graph-Based Dependency Parsing.” ACLWeb. Gyeongju, Republic of Korea: International Committee on Computational Linguistics. October 1, 2022. https://aclanthology.org/2022.coling-1.483/.

[5] Zhang, Yu, Zhenghua Li, and Min Zhang. 2020. “Efficient Second-Order TreeCRF for Neural Dependency Parsing.” ACLWeb. Online: Association for Computational Linguistics. July 1, 2020. https://doi.org/10.18653/v1/2020.acl-main.302.

Model overview



An overview of our model. We take the idea from [2] and replace the LSTM with next-generation encoders