

1. Phương pháp

1.1 Mô hình Graph-based Dependency Parser

Nhóm sử dụng mô hình [Stanford's Graph-based Neural Dependency Parser at the CoNLL 2017 Shared Task](#) để xây dựng mô hình Dependency Parser.

Nhóm tập trung vào thử nghiệm các bộ embedding khác nhau trong 4 hệ thống

	Embeddings
Mô hình 1	Embedding Layer + Character-level Embeddings
Mô hình 2	Embedding Layer + Tag Embedding
Mô hình 3	Pretrained Word Embedding + Character-level Embeddings
Mô hình 4	Pretrained Word Embedding + Tag Embedding

- Embedding Layer được áp dụng cho mức từ, với kích thước 400
- Character-level Embeddings được áp dụng cho mức character, với kích thước 300
- Pretrained Word Embeddings sử dụng baomoi.model.bin từ <https://github.com/sonvx/word2vecVN>
- Tag Embedding được áp dụng cho POS Tag, với kích thước 300.

Các tham số mặc định của mô hình Neural Dependency Parser

BiLSTM Layers	#number layer: 3 # output layer size: 400 # dropout 0.33
Attention Layers	(mlp_arc_d) <ul style="list-style-type: none">• MLP(n_in=800, n_out=500, dropout=0.33) (mlp_arc_h) <ul style="list-style-type: none">• MLP(n_in=800, n_out=500, dropout=0.33) (mlp_rel_d) <ul style="list-style-type: none">• MLP(n_in=800, n_out=100, dropout=0.33) (mlp_rel_h) <ul style="list-style-type: none">• MLP(n_in=800, n_out=100, dropout=0.33)

	(arc_attn) <ul style="list-style-type: none"> Biaffine(n_in=500, n_out=1) (rel_attn): <ul style="list-style-type: none"> Biaffine(n_in=100, n_out=172)
--	--

1.2 Xử lý dữ liệu, huấn luyện mô hình

Từ dữ liệu của ban tổ chức, nhóm gộp toàn bộ tập dữ liệu từ VTB, HTB và MXH, lọc các câu trùng được bộ dữ liệu 7454 câu.

Dữ liệu này được chia thành 3 phần train (6499 câu), dev (499 câu) và test (453 câu). Trích xuất các câu này thành hai bộ dữ liệu: một bộ dữ liệu để huấn luyện mô hình POS Tag sử dụng nhãn UPOS, một bộ dữ liệu để huấn luyện mô hình Dependency Parser.

- Bộ dữ liệu mô hình POS Tag được huấn luyện theo mô hình Bidirectional LSTM-CRF Models for Sequence Tagging ở [Huang et al. 2015](#) với bộ pre-trained word embeddings sử dụng baomoi.model.bin (kích thước 400) để xây dựng POS Tagger.
- Bộ dữ liệu Dependency Parser để xây dựng mô hình dependency parser sử dụng phương pháp đã mô tả kể trên.

Đối với dữ liệu raw, nhóm sử dụng [underthesea](#) để tách từ, sử dụng bộ POS Tagger huấn luyện để tạo ra các nhãn POS Tag, sau đó được đưa vào hai mô hình dependency parser được mô tả trong mô hình 1 và mô hình 2.

Đối với dữ liệu định dạng conll, nhóm đưa vào mô hình dependency parser được mô tả trong mô hình 1, 2, 3 và 4.

Các mô hình được huấn luyện trên máy với cấu hình Intel® Core™ i7-8700, 32 GB Ram, Card GeForce GTX 1080 Ti. Thời gian huấn luyện mô hình POS Tagger là 32 phút, thời gian huấn luyện mô hình Dependency Parser trung bình với mỗi mô hình là 64 phút.

2. Kết quả

2.1 Kết quả trong quá trình huấn luyện

Kết quả huấn luyện mô hình POS Tagger trên tập dev/test

	F1 (score)
Dev	96.2%
Test	95.8%

Kết quả huấn luyện mô hình Dependency Parser trên tập dev/test

	Dev		Test	
	UAS	LAS	UAS	LAS
Mô hình 1	79.18%	67.25%	80.90%	68.52%
Mô hình 2	78.72%	66.28%	79.23%	67.24%
Mô hình 3	80.21%	68.23%	80.47%	69.01%
Mô hình 4	80.82%	69.45%	81.02%	69.75%

2.2 Kết quả đối với tập private test từ ban tổ chức

Đối với dữ liệu raw text

	UAS	LAS
Mô hình 1	75.63%	67.12%
Mô hình 2	74.15%	64.93%

Đối với dữ liệu CONLL

	UAS	LAS
Mô hình 1	79.86%	70.62%
Mô hình 2	80.81%	72.66%
Mô hình 3	80.44%	71.5%

Mô hình 4	81.53%	72.96%
-----------	---------------	---------------

3. Kết luận

Đối với dữ liệu raw text, việc sử dụng character-level embeddings có kết quả tốt hơn so với sử dụng POS Tag Embedding.

Đối với dữ liệu CoNLL, việc sử dụng pretrained word embeddings kết hợp với POS Tag Embeddings (với gold label) cho kết quả tốt nhất.

Hướng nghiên cứu tiếp của nhóm là sử dụng PhoBert Embedding, và thử nghiệm thêm các mô hình mạng neural network cho dependency parser sử dụng như đề xuất ở paper [Efficient Second-Order {Tree}{CRF} for Neural Dependency Parsing](#)