다양한 앙상블 알고리즘을 이용한 한국어 의존 구문 분석

Korean Dependency Parsing Using Various Ensemble Models

조경철, 김주완, 김균엽, 박성진, 강상우

가천대학교, 소프트웨어학과

cjf9028@daum.net, rlawndhks217@gmail.com, gyop0817@naver.com,
parksjin01@gmail.com, swkang@gmail.com

Department of Software, Gachon University

요 약

본 논문은 최신 한국어 의존 구문 분석 모델(Korean dependency parsing model)들과 다양한 앙상블 모델(ensemble model)들을 결합하여 그 성능을 분석한다. 단어 표현은 미리 학습된 워드 임베딩 모델(word embedding model)과 ELMo(Embedding from Language Model), Bert(Bidirectional Encoder Representations from Transformer) 그리고 다양한 추가 자질들을 사용한다. 또한 사용된 의존 구문 분석 모델로는 Stack Pointer Network Model, Deep Biaffine Attention Parser와 Left to Right Pointer Parser를 이용한다. 최종적으로 각 모델의 분석 결과를 앙상블 모델인 Bagging 기법과 XGBoost(Extreme Gradient Boosting) 이용하여 최적의 모델을 제안한다.

주제어: 의존 구문분석, 전이 기반 구문분석, ELMo, Bert, XGBoost

1. 서론

구문 분석(parser)은 문장의 구조를 이해하고 의미를 파악해 내는 과정이다. 의존 구문 분석(dependency parser)은 문장 내의 성분들을 지배 의존 관계로 파악하고 구문 분석의 복잡도를 줄임으로서 효과적으로 구문을 분석한다.

최근 의존 구문 분석을 위하여 기계 학습을 사용한 연구가 활발히 진행되고 있으며 모델의 입력을 위한 단어 표현 기법으로 최근 문맥 정보를 반영한 워드 임베딩 모델들이 주목받고 있다. 본 논문에서는 단어 표현을 위하여 ELMo[1]와 Bert[2]를 사용하고 최신 의존 구문 파싱 모델들의 다양한 조합을 적용한다. 본 논문에서는 Stack Pointer Network Model[3][4], Deep Biaffine Attention parser[5], Left2Right Pointer Parser[6] 기법들을 사용하고 다양한 단어 표현을 조합하여 모델 간 성능을 분석하고 활용 가능성을 제시한다.

앙상블 모델은 단일 기계 학습을 보완하기 위한 방법으로써 다양한 모델의 결과들을 조합해 더 높은 정확도를 도출한다고 알려져 있다. 본 논문에서는 Bagging, XGBoost을 이용한 다양한 앙상블 모델들을 적용하여 최적의 모델을 제안한다.

2. 관련 연구

최근 자연어 처리 분야 연구에 기계 학습의 한 방법인 DNN(Deep Neural Network)을 이용하는 방법들이 제안되고 있다. 의존 구문 분석 분야에서는 전이 기반 방법의 연구가 활발하게 연구되고 있다. 전이 기반 모델은 학습을 통해 전이 히스토리에서 다음 상태 전이를 예상하기 위한 모델을 만들고 구문 분석 단계에서 학습을 통해 만들어진 모델로부터 최상의 전이열을 생성한다. 이 방식은 결정론적 의존 구문 분석 방법이며 greedy 알고리즘을 기반으로 한 지역적 학습 모델을 사용한다. Stack Pointer Network Model은 전이 기반 구조의 대표적인 모델로 attention 메커니즘을 이용하여 입력 문장에서 각 어절의 핵심어 위치와 의존 레이블 정보를 출력하여 의존관계를 파악한다.

최근 의존 구문 분석을 위한 다양한 단어 표현 방법이 꾸준히 연구되어 왔다. 기존의 워드 임베딩은 Word2Vec와 FastText 등이 있는데 기존의 방법들은 단어 벡터들이 하나의 고정된 값만을 가지는 문제점이 있다. 언어모델을 이용하여 학습시킨 Elmo와 Bert는 문맥에 따라서 같은 단어여도 다른 단어 벡터를 가지게 되어 기존의 워드 임베딩의 문제점을 해결하였다. Elmo는 Bi-LSTM(Bidirectional Long Short-Term Memory Models)을 이용하여 양방향으로 학습시킨 언어모델이며 Bert는 attention 으로 구성된 Transformer를 이용하여 학습시킨 언어모델이다.

3. 제안 모델

3.1 데이터 전처리

학습 데이터의 정제를 위하여 다음과 같은 과정들을 진행하였다. 한자는 의미와 형태에 관계없이 '/SH' 형태소 태그(POS tag)를 갖는다. 동일한 의미의 한자와 한글을 동일한 단어로 만들어 주기 위해 한중일 통합 및 호환용 한자 사전을 이용하여 한자를 한글로 변환하는 과정을 추가하였다. 또한 구두점이나 심볼들이 연속되거나 부정확하게 사용되는 경우 해당 표현을 수정하였고 줄바꿈이나 띄어쓰기 등의 잘못되거나 중복되는 오류를 제거하였다. 테스트 데이터에서 관측되는 unknown 단어와 형태소들은 훈련 데이터 중에서 형태적으로 유사한 단어 혹은 형태소로 대치하였다.

3.2 공통 자질

제안하는 모델을 위해 입력 문장을 CoNLL-U[7]의 형태로 변환하는 과정이 필요하다. 입력 문장을 형태소 분석을 수행하고 어절 단위로 구분한 뒤 어절의 첫 형태소와 두 번째 형태소, 마지막 이전 형태소 그리고 마지막 형태소를 하나의 심볼로 결합하여 해당 어절의 품사 자질로 사용한다. 한 어절이 한 개의 형태소로 구성되어있다면 하나의 형태소를 하나의 심볼로 사용하고 두 개의 형태소인 경우 두 개의 형태소를 하나의 심볼로 결합한다.

워드 임베딩은 단어를 특정 차원의 실수형 벡터로 표현하는 것을 의미한다. 본 논문에서는 임베딩 벡터의 차원을 Elmo의 경우 1024차원으로 Bert의 경우 768 차원으로 설정하여 미리 학습된 50차원 벡터와 같이 사용하였다.

3.3 구문 분석을 위한 앙상블 모델

본 논문에서는 Stack Pointer Network Model, Deep Biaffine Attention Parser, Left2Right Pointer Parser를 이용하고 Bagging, XGBoost[8] 을 이용하여 앙상블 모델을 구성한다. 제안한 앙상블 모델은 각 분석 모델들의 결과들을 종합하여 여러 알고리즘을 이용해 결과를 생성한다.

그림 1은 배깅 기법의 흐름도를 보여준다. 각 모델의 가중치를 $w_1 w_{2_} w_n$으로, 각 모델의 결과 값들을 $x_1 x_{2\ldots} x_n$으로 나타내면 각 모델의 결과 값은 $w_n x_n$이 된다. 앙상블 모델의 결과 값은 max($w_n x_n$)이 선택된다.

또한 XGBoost 기법을 이용한 앙상블 모델들은 모든 모델의 결과 값을 Arc 와 Label을 나누어 학습을 시켰다. Arc의 경우 모델의 결과 값을 상대적인 위치로 변환하였으며 Label의 경우 Label을 인덱스로 변환하여 학습을 진행한다.

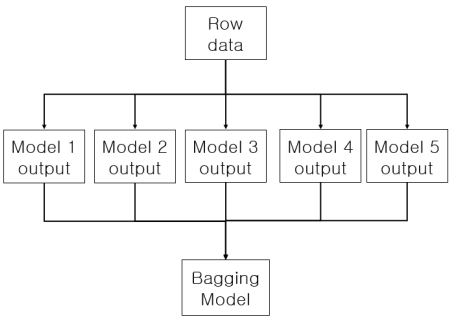

그림 1 배깅 기법의 흐름도

3.3.1 Stack Pointer Network Model

Stack Pointer Network Model은 attention 메커니즘을 기반으로 입력 열의 위치가 출력되며 입력열의 위치에 대한 확률 분포를 학습하는 모델이다. Bi-LSTM, 인코더(encoder)-디코더(decoder)모델을 기반으로 하며 hidden state를 통해 attention 가중치를 학습하고 의존 레이블과 의존 관계에 있는 단어의 위치를 출력한다.

Stack Pointer Network Model은 형제노드와 조부모 노드, 중심어를 조합하여 디코더의 입력으로 사용한다.

3.3.2 Deep Biaffine Attenion Parser

Arc를 결정하는 부분과 Label을 결정하는 부분으로 구성이 된다. Bi-LSTM을 적용시켜 단어 표상을 얻고 MLP를 적용시켜 2개의 추상화된 표상을 얻는다. Biaffine Attention을 수행하여 각 단어의 head를 결정한다. 별도의 MLP를 적용시켜 2개의 표상을 얻고 Biaffine 변환과 결합된 함수를 이용하여 의존관계를 결정한다.

3.3.3 Left to Right Pointer Parser

Stack Pointer Network와 동일하게 attention 메커니즘을 기반으로 입력 열의 위치가 출력되며 입력열의 위치에 대한 확률 분포를 학습하는 모델이다. LSTM, 인코더-디코더 모델을 기반으로 하며 hidden state를 통해 attention 가중치를 학습하고 의존 레이블과 의존 관계에 있는 단어의 위치를 출력한다.

디코더의 입력으로는 Stack Pointer Network와는 다르게 중심어의 단어만 사용한다.

4. 실험 및 결과

실험에 사용된 말뭉치는 의존 구조로 변환된 세종 데이터[9] 셋을 사용했으며 데이터 셋은 총 59,737 문장이며 이 중 약 90%인 53,920문장을 학습에 사용하고 약 10%인 5,817문장을 평가에 사용하였다.

앙상블 모델의 훈련을 위해서는 각 파싱 모델의 출력 값을 사용하였으며 약 57,954 어절을 포함한다. 또한 약 10%인 6,000 어절을 평가에 사용하였다.

단일 모델의 성능 분석을 위하여 Stack Pointer Network Model, Deep Biaffine Attention Parser, Left2Right Pointer Parser의 성능 차이를 비교 평가 하였다. 학습된 벡터는 50차원으로 고정하고 ELMo 와 Bert 를 적용시켜 성능을 비교하였다. 평가 척도로는 UAS(Unlabeled Attachment Score)와 LAS(Labeled Attachment Score)는 식 1과 2를 사용하여 평가하였다.

$$UAS = \frac{\text{구문 분석 의존 트리에서의 모든 아크수}}{\text{정답 의존 트리에서의 모든 아크 수}} \quad (1)$$

$$LAS = \frac{\text{구문 분석 의존 트리에서의 모든 아크수 + 레이블 수}}{\text{정답 의존 트리에서의 모든 아크 수 + 레이블 수}} \quad (2)$$

표 1 제안한 한국어 의존 구문 분석 모델의 성능 비교

모델	정확도	
	UAS	LAS
Stack Pointer Network Model+ Elmo	91.01	89.18
Stack Pointer Network Model+ Bert	91.39	89.28
Deep Biaffine Attention Parser+ ELMo	91.35	88.80
Deep Biaffine Attention Parser+ Bert	**92.01**	**89.89**
Left to Right Pointer Parser + ELMo	91.73	89.32
Left to Right Pointer Parser + Bert	91.46	89.13

표 1은 ELMo와 Bert를 이용한 한국어 의존 구문 분석의 실험 결과를 나타낸다. Deep Biaffine Model에 Bert를 사용한 것이 UAS 92.06%, LAS 90.62%로 가장 높은 성능을 보였다.

표 2 단일 모델과 앙상블 모델 성능 비교

앙상블 모델	UAS	LAS
Ensemble model (Bagging)	92.30	90.20
Ensemble model (XGBoost)	**94.44**	**94.1**
Deep Biaffine Attention Parser+ Bert	92.01	89.89

표 3은 구문 분석 모델 중 가장 높은 성능을 보여준 Deep Biaffine Attention Parser (Bert)과 앙상블 모델들의 성능을 비교한 결과이다. XGBoost 기법을 이용한 앙상블 모델에서 UAS 2.4%, LAS 4.2%의 성능이 향상됨을 보여준다.

5. 결론

제안한 모델은 최신의 구문 분석 모델들의 결과들을 종합하여 3가지 앙상블 모델을 구성하였으며 입력 자질을 위하여 ELMo와 Bert 그리고 다양한 추가 자질들을 사용하였다. 실험을 통하여 제안한 모델들의 성능을 비교하였으며 XGBoost 기법을 이용한 앙상블 모델을 통하여 단일 모델인 경우와 비교하여 UAS, LAS에서 각각 2.4%p, 4.2%p의 성능이 향상되었음을 증명하였다.

감사의 글

이 연구는 2019년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2019R1C1C1006299)

참고문헌

[1] M. Peters, M. Neumann, M. Iyyer, M. Gardner, “Deep contextualized word representations”, NAACL, 2018.
[2] J.Devlin, M. Chang, K. Lee and K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2018)
[3] 박천음, 이창기 “포지션 인코딩 기반 스택 포인터 네트워크를 이용한 한국어 상호 참조 해결”, 정보과학회논문지 제24권 제3호, p.113-121, 2018.3
[4] Xuezhe Ma, Zecong Hu, Jingzhou Liu, Nanyun Peng, Graham Neubig, and Eduard Hovy. “Stackpointer networks for dependency parsing.” ACL, 2018.
[5] T. Dozat, C. Manning. “Deep Biaffine Attention for Neural Dependency Parsing”, ICLR, 2017.
[6] Daniel Fernández-González,Carlos Gómez-Rodríguez “Left-to-Right Dependency Parsing with Pointer Networks”ACL p. 710-716, 2019
[7] S. Bucholz, E. Marsi, CoNLL-X shared task on Multilingual Dependency Parsing, Proc. of CoNLL, p.149-164, 2006.
[8] Liu, G., Nguyen, T. T., Zhao, G., Zha, W., Yang, J., Cao, J. & Chen, W. “Repeat buyer prediction for e-commerce”, In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining ACM. pp. 155-164, 2016
[9] Century Sejong Plan, 2012 (in Korean)