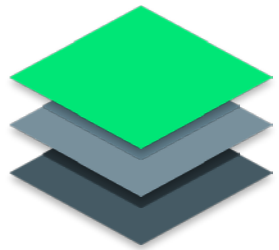
Dependency Parsing of Korean using SyntaxNet

고려대학교 컴퓨터학과
Andrew Matteson
Web : www.andrewmatteson.name
E-mail : amatteson@korea.ac.kr



Deep Learning for NLP lecture

고려대학교 정보대학 컴퓨터학과

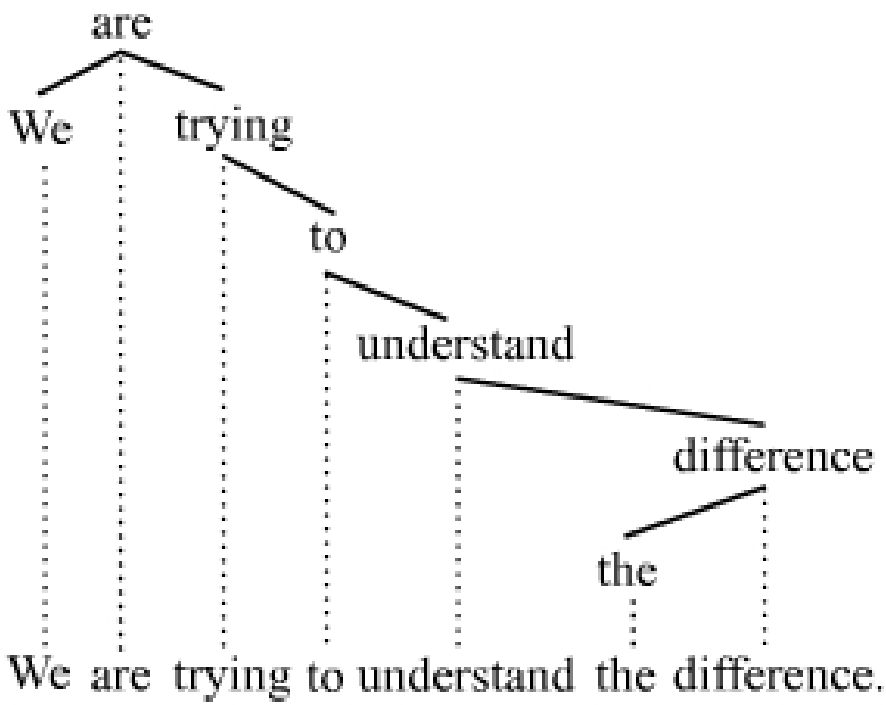
2017

Index

Introduction

Dependency Grammar

- Analyze dependencies between certain words



Dependency

SyntaxNet (1/3)

- Package by Google allowing:
  - Tokenization
  - Morphological analysis
  - POS tagging
  - Dependency parsing
- Only supports Python 2

SyntaxNet (2/3)

- Embeds sets of features deeply
- Hidden layer of words can also serve as word vector representation
- Two network graph structures are provided:
  - **Greedy:** picks only best #1 choice at each step of parsing
  - **Structured:** allows beam search with Conditional Random Field objective over several hypotheses

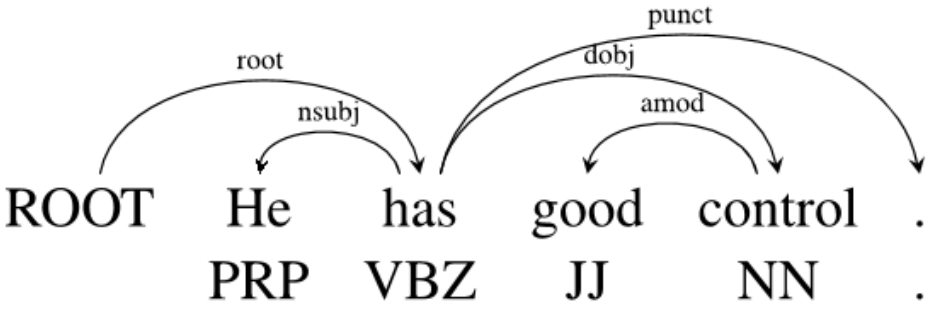
SyntaxNet (3/3)

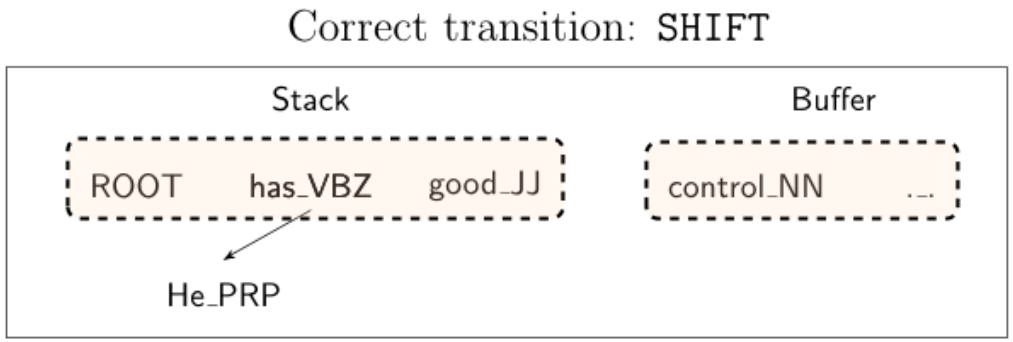
- Information typically flows from top (tokenizer) to bottom (dependency parser)
  - Output of tagging used as input to parsing
- “Parsey McParseface”

SyntaxNet Parser

- Arc-standard transition system
- Input sentences
  - Represented as a sequence of transition actions and labels
- Train input parser state against gold standard output parser state

Arc-standard Transition System





Transition	Stack	Buffer	A
	[ROOT]	[He has good control .]	$\emptyset$
SHIFT	[ROOT He]	[has good control .]	
SHIFT	[ROOT He has]	[good control .]	
LEFT-ARC(nsubj)	[ROOT has]	[good control .]	$A \cup$ nsubj(has,He)
SHIFT	[ROOT has good]	[control .]	
SHIFT	[ROOT has good control]	[.]	
LEFT-ARC(amod)	[ROOT has control]	[.]	$A \cup$amod(control,good)
RIGHT-ARC(dobj)	[ROOT has]	[.]	$A \cup$ dobj(has,control)
...	...	...	...
RIGHT-ARC(root)	[ROOT]	[]	$A \cup$ root(ROOT,has)

Chen, Manning (EMNLP 2014)

Challenges

Challenges of SyntaxNet

- Lack of documentation
- Input features to parser are important
- Processes sentences as a series of tokens
  - Tokens assumed to be separated by whitespace
  - Separating Korean into tokens is difficult
- Can’t yet use SyntaxNet tagger as input to parser in Korean case
  - Tagging Korean in SyntaxNet difficult due to lack of Korean features
- No GPU support (slow)

Challenges of Korean

- Morphologically complex
- Some markers in Korean are optional
  - 나는 닭발을 잘 안 먹어
  - 나 닭발 잘 안 먹어
- Honorifics/verb stem changes
  - 저는 닭발을 잘 안 먹어요
  - 우리 교수님은 닭발을 잘 안 드세요

Preparation

Prerequisites

- Sejong Written Language Corpus (Constituency-Tree Format)
  - “현대문어 구문분석 말뭉치” available on sejong.or.kr
  - Preprocessing/output to CoNLL-U Format
- Google SyntaxNet
  - Installation process is well-documented
- Third-party POS tagger
  - Not using SyntaxNet for tagging
- Custom scripts
  - Convert between necessary formats, etc…

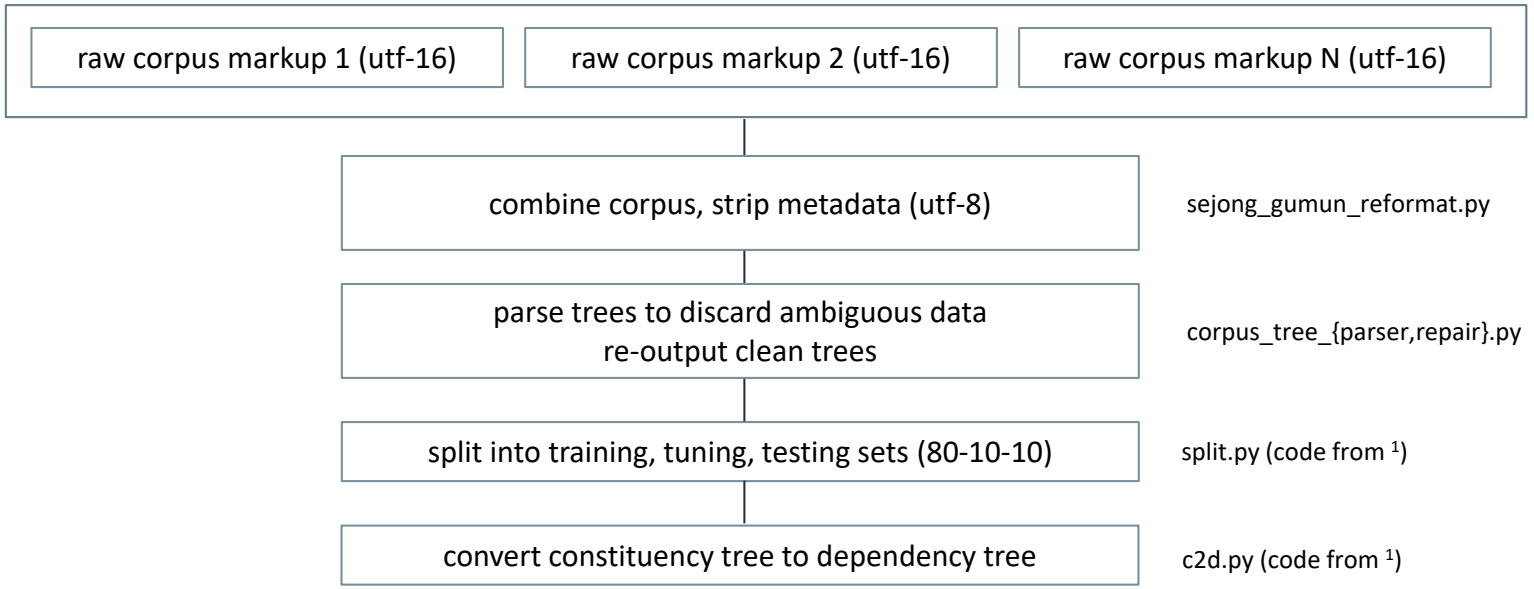
Sejong Corpus (1/5)

- Available freely online under Miscellaneous section of sejong.or.kr
- Sample sentence:

```
; 자신도 고교를 졸업하고 용돈을 벌 용량으로 띄엄띄엄 공사판을 돌아다니다 배관 기술을 익혀 눌러 앉았다는 전씨의 월수입은 1백 40여 만원.
(S      (NP_SBJ         (NP_MOD         (S_MOD  (NP_SBJ 자신/NNG + 도/JX)
                                (VP_MOD         (VP     (VP     (VP     (NP_OBJ 고교/NNG + 를/JKO)
                                                                (VP 졸업/NNG + 하/XSV + 고/EC))
                                                        (VP     (NP_AJT         (VP_MOD         (NP_OBJ 용돈/NNG + 을/JKO)
                                                                                (VP_MOD 벌/VV + ㄹ/ETM))
                                                                        (NP_AJT 용량/NNG + 으로/JKB))
                                                                (VP     (AP 띄엄띄엄/MAG)
                                                                        (VP     (NP_OBJ 공사판/NNG + 을/JKO)
                                                                                (VP 돌아다니/VV + 다/EC)))))
                                                        (VP     (NP_OBJ         (NP 배관/NNG)
                                                                        (NP_OBJ 기술/NNG + 을/JKO))
                                                                (VP 익히/VV + 어/EC)))
                                        (VP_MOD         (VP 누르/VV + 어/EC)
                                                (VP_MOD 앉/VV + 았/EP + 다는/ETM))))
                        (NP_MOD 전/NNP + 씨/NNB + 의/JKG))
                (NP_SBJ 월/NNG + 수입/NNG + 은/JX))
        (NP     (NP     (NP 1/SN + 백/NR)
                        (NP 40/SN + 여/XSN))
                (NP 만/NR + 원/NNG + ./SF)))
```

Sejong Corpus (2/5)

Overview



[1] https://github.com/dsindex/syntaxnet

Sejong Corpus (3/5)

Result

```
<!DOCTYPE tei.2 SYSTEM "c:\sgml\dtd\tei2.dtd" [
        <!ENTITY % TEI.corpus "INCLUDE">
        <!ENTITY % TEI.extensions.ent SYSTEM "sejong1.ent">
        <!ENTITY % TEI.extensions.dtd SYSTEM "sejong1.dtd">
]>

<tei.2>
<teiHeader>
        <fileDesc>
                <titleStmt>
                        <title>조선일보 생활(93),  구문 분석 전자파일</title>
                        <author>조선일보사</author>
                        <sponsor>대한민국 문화관광부</sponsor>
                        <respStmt>
                                <resp>문헌입력, 표준화, 구문 정보 부착</resp>
…
<body>
; 1993/06/08 19
(NP     (NP 1993/SN + //SP + 06/SN + //SP + 08/SN)
        (NP 19/SN))
```

```
; 1993/06/08 19
(NP     (NP 1993/SN + //SP + 06/SN + //SP + 08/SN)
        (NP 19/SN))

; 엠마누엘 웅가로 /
(NP     (NP     (NP 엠마누엘/NNP)
                (NP 웅가로/NNP))
        (X //SP))

; 의상서 실내 장식품으로…
(NP_AJT         (NP_AJT 의상/NNG + 서/JKB)
        (NP_AJT         (NP 실내/NNG)
                (NP_AJT 장식품/NNG + 으로/JKB + …/SE)))

; 디자인 세계 넓혀
(VP     (NP_OBJ         (NP 디자인/NNG)
                (NP_OBJ 세계/NNG))
        (VP 넓히/VV + 어/EC))
```

Sejong Corpus (4/5)

- Depth-first traversal of constituency tree
- Output to CoNLL-U format (introduced later)
- Determine HEAD of all leaf nodes

Sejong Corpus (5/5)

Constituency Tree -> Dependency Tree (c2d)

```
; 프랑스의 세계적인 의상 디자이너 엠마누엘 웅가로가 실내 장식용 직물 디자이
너로 나섰다.
(S      (NP_SBJ     (NP     (NP_MOD 프랑스/NNP + 의/JKG)
            (NP     (VNP_MOD 세계/NNG + 적/XSN + 이/VCP + ㄴ/ETM)
                (NP     (NP 의상/NNG)
                    (NP 디자이너/NNG))))
        (NP_SBJ     (NP 엠마누엘/NNP)
            (NP_SBJ 웅가로/NNP + 가/JKS)))
    (VP     (NP_AJT     (NP     (NP     (NP 실내/NNG)
                    (NP 장식/NNG + 용/XSN))
                (NP 직물/NNG))
            (NP_AJT 디자이너/NNG + 로/JKB))
        (VP 나서/VV + 었/EP + 다/EF + ./SF)))
```

```
1       프랑스의        프랑스/NNP + 의/JKG     NP_MOD  4
2       세계적인        세계/NNG + 적/XSN + 이/VCP + ㄴ/ETM      VNP_MOD 4
3       의상    의상/NNG        NP      4
4       디자이너        디자이너/NNG    NP      6
5       엠마누엘        엠마누엘/NNP    NP      6
6       웅가로가        웅가로/NNP + 가/JKS     NP_SBJ  11
7       실내    실내/NNG        NP      8
8       장식용  장식/NNG + 용/XSN       NP      9
9       직물    직물/NNG        NP      10
10      디자이너로      디자이너/NNG + 로/JKB   NP_AJT  11
11      나섰다. 나서/VV + 었/EP + 다/EF + ./SF  VP      0
```

Credits to Myungchul Shin[1]

[1] https://github.com/dsindex/syntaxnet

Google SyntaxNet

- https://github.com/tensorflow/models/tree/master/syntaxnet
- As of writing, Python 2.7 only
- Install prerequisites for SyntaxNet
- Compilation process hogs huge amount of RAM and CPU
- Compile and verify SyntaxNet tests succeed

POS Tagger (1/2)

- Reassembly requires POS tagger that supports 어절 indexing
  - Reassembly makes output human-readable
  - Provides index of input character in morphological analysis
- Unaware of open source tagger providing this
  - Except Komoran 3.0 beta

POS Tagger (2/2)

- Komoran 3.0 Beta
  - http://shineware.tistory.com/entry/KOMORAN-30-beta
  - https://github.com/shin285/KOMORAN
- Requires dependency jar files by same author
- Compile in Eclipse Java workspace

Custom Scripts

- Myungchul Shin's github
  - https://github.com/dsindex/syntaxnet
  - split.py : 80/10/10 training/tuning/testing
  - c2d.py : constituency tree -> dependency tree
- Remove erroneous sentences from corpus
  - Until c2d.py succeeds
  - Manually or write script
- Reassembly of components to Eojeol (optional)
  - Use token 어절 index from POS tagger
- Tree visualization
  - CoNLL-U -> tree
  - SVG file output

Additional Info

- Read available documentation

- Google SyntaxNet
  - Training process/etc.
  - https://github.com/tensorflow/models/tree/master/syntaxnet

- Myungchul Shin's github
  - Sejong corpus related info
  - https://github.com/dsindex/syntaxnet

Training

SyntaxNet Configuration (1/2)

- Input corpus needs to be in CoNLL-U Format

ID	FORM	LEMMA	UPOSTAG	XPOSTAG	FEATS	HEAD	DEPREL	DEPS	MISC
1	나	_	NP	NP	_	2	MOD	_	_
2	는	_	JX	JX	_	5	NP_SBJ	_	_
3	자연언어처리	_	NNP	NNP	_	4	MOD	_	_
4	를	_	JKO	JKO	_	5	NP_OBJ	_	_
5	좋아하	_	VV	VV	_	6	MOD	_	_
6	ㄴ다	_	EF	EF	_	7	MOD	_	_
7	.	_	SF	SF	_	0	ROOT	_	_

SyntaxNet Configuration (2/2)

- Configuration file called context.pbtxt
- Configures parser
- Designates input features to parser

This Model's Features (1/3)

`token.word:` word embedding for specified token

Feature Group: words (dim=64)	
input.word	stack(1).child(1).word
input(1).word	stack(1).child(1).sibling(-1).word
input(2).word	stack(1).child(-1).word
input(3).word	stack(1).child(-1).sibling(1).word
stack.word	stack.child(2).word
stack(1).word	stack.child(-2).word
stack(2).word	stack(1).child(2).word
stack(3).word	stack(1).child(-2).word
stack.child(1).word	
stack.child(1).sibling(-1).word	
stack.child(-1).word	
stack.child(-1).sibling(1).word	

This Model's Features (2/3)

`token.tag:` POS tag used in specified token (previous, current, or future tokens)

Feature Group: tags (dim=32)	
input.tag	stack(1).child(1).tag
input(1).tag	stack(1).child(1).sibling(-1).tag
input(2).tag	stack(1).child(-1).tag
input(3).tag	stack(1).child(-1).sibling(1).tag
stack.tag	stack.child(2).tag
stack(1).tag	stack.child(-2).tag
stack(2).tag	stack(1).child(2).tag
stack(3).tag	stack(1).child(-2).tag
stack.child(1).tag	
stack.child(1).sibling(-1).tag	
stack.child(-1).tag	
stack.child(-1).sibling(1).tag	

This Model's Features (3/3)

`token.label:` predicted DEPREL label (stack (previously processed) tokens only)

Feature Group: labels (dim=32)
stack.child(1).label
stack.child(1).sibling(-1).label
stack.child(-1).label
stack.child(-1).sibling(1).label
stack(1).child(1).label
stack(1).child(1).sibling(-1).label
stack(1).child(-1).label
stack(1).child(-1).sibling(1).label
stack.child(2).label
stack.child(-2).label
stack(1).child(2).label
stack(1).child(-2).label

SyntaxNet Training Script

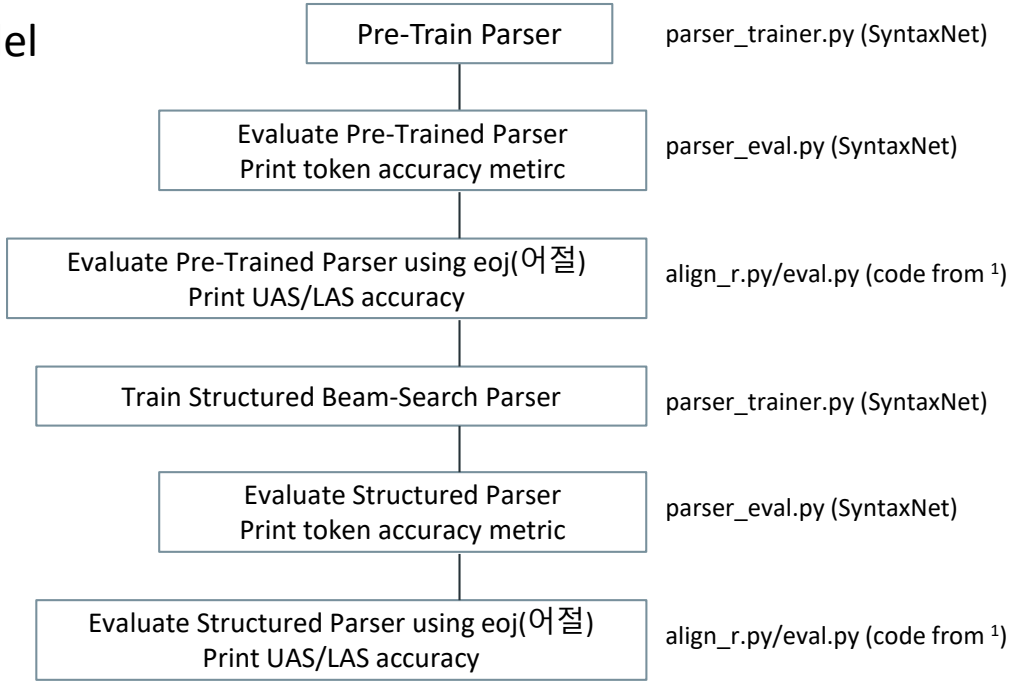
Training Script

Bash script to train SyntaxNet model

```
GP_PARAMS=${HIDDEN_LAYER_PARAMS}-0.02-100-0.9
function train_parser {
        ${BINDIR}/parser_trainer \
          --arg_prefix=brain_parser \
          --batch_size=${BATCH_SIZE} \
          --compute_lexicon \
          --decay_steps=100 \
          --graph_builder=structured \
          --hidden_layer_sizes=${HIDDEN_LAYER_SIZES} \
          --learning_rate=0.02 \
          --momentum=0.9 \
          --beam_size=${BEAM_SIZE} \
          --output_path=${TMP_DIR} \
          --task_context=${TMP_DIR}/brain_parser/greedy/${LP_P
          --training_corpus=projectivized-training-corpus \
          --tuning_corpus=tagged-tuning-corpus \
          --params=${GP_PARAMS} \
          --pretrained_params=${TMP_DIR}/brain_parser/greedy/$
          --pretrained_params_names=embedding_matrix_0,embeddi
          --num_epochs=10 \
          --report_every=25 \
          --checkpoint_every=200 \
          --logtostderr
}
```

…

```
pretrain_parser
evaluate_pretrained_parser
evaluate_pretrained_parser_by_eoj
train_parser
evaluate_parser
evaluate_parser_by_eoj
copy_model
```



Label Attachment Score (LAS) – % of tokens with correct HEAD and DEPREL
Unlabeled Attachment Score (UAS) – % of tokens with correct HEAD

[1] https://github.com/dsindex/syntaxnet

Greedy Training

- Required first step
- Usable model can be generated in just a couple hours

Greedy Training - Hyperparameters

- Batch size: 256 [number of individual tokens (incl. all features) input at once]
- Hidden layers:
  - 512 (ReLU activation)
  - 512 (ReLU activation)
- Output layer:
  - Softmax activation for choosing best transition action
- Objective: Minimize binary cross-entropy
- Optimizer: tf.train.MomentumOptimizer
  - Learning rate: 0.08
  - Momentum: 0.85
- Decay steps: 4400 [decay learning rate by 0.96 every *n* steps]
- Number of epochs: 20 [number of full passes through entire training set]

Structured Training

- Requires greedy model
- Uses beam search
- Identifies and fixes training faults in greedy model
- Can take a day or two

Structured Training - Hyperparameters

- Batch size: 256
- Hidden layers:
  - 512 (ReLU activation)
  - 512 (ReLU activation)
- Output layer:
  - Softmax activation for choosing best transition action
  - Beam search also performed
- Objective: Minimize binary cross-entropy
- Optimizer: tf.train.MomentumOptimizer
  - Learning rate: 0.02
  - Momentum: 0.90
- Beam size: 16
- Decay steps: 100
- Number of epochs: 10

Accuracy Assessment

- **Label Attachment Score (LAS)** - % of tokens with correct HEAD and DEPREL
- **Unlabeled Attachment Score (UAS)** - % of tokens with correct HEAD

Evaluated Sentence Count	
Training	Testing
19,569	3,788

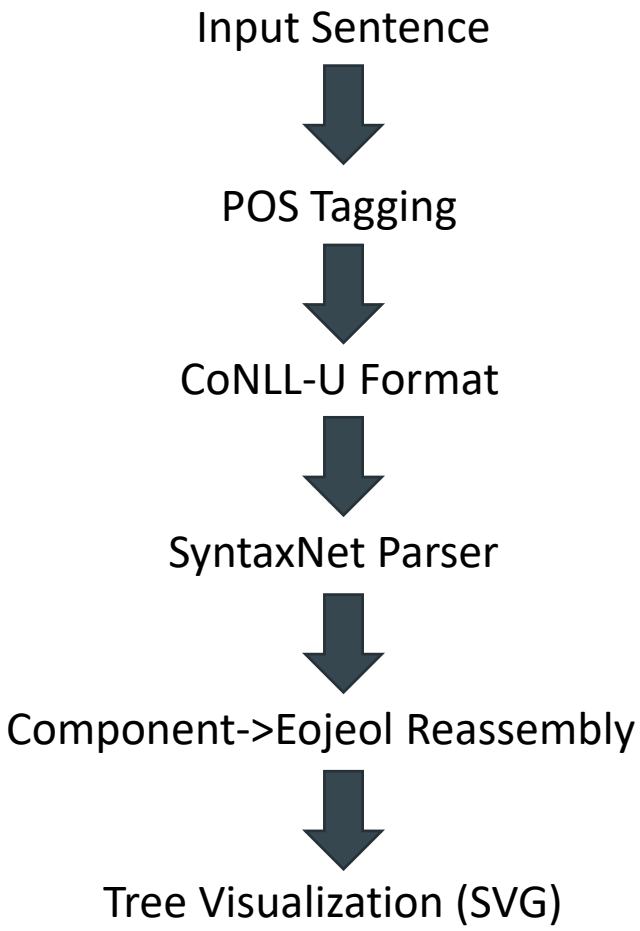
	UAS		LAS	
	Training	Testing	Training	Testing
Greedy	92.17%	87.71%	89.56%	84.58%
Structured	**94.61%**	**88.68%**	**92.13%**	**85.13%**

Execution

Execution

- So far, we have a trained model
- But running model is a whole different process

Workflow


Input Sentence
POS Tagging
CoNLL-U Format
SyntaxNet Parser
Component->Eojeol Reassembly
Tree Visualization (SVG)

POS Tagging (1/2)

- Invoke Komoran 3 on input sentence (Java)
- Save 어절 index of each token for later reassembly
  - Can be saved in CoNLL-U metadata field or externally
- Output to CoNLL-U Format
  - Only fill in POS tagging fields
  - HEAD or DEPREL later filled by SyntaxNet parser

POS Tagging (2/2)

- Input: “… 회사로 분할하기로 한 것은 …”
- Part of speech tagging

ID	FORM	LEMMA	UPOSTAG	XPOSTAG	FEATS	HEAD	DEPREL	DEPS
11	회사	회사	NNG	NNG	_	_	_	_
12	로	로	JKB	JKB	_	_	_	_
13	분할	분할	NNG	NNG	_	_	_	_
14	하	하	XSV	XSV	_	_	_	_
15	기	기	ETN	ETN	_	_	_	_
16	로	로	JKB	JKB	_	_	_	_
17	한	한	MM	MM	_	_	_	_
18	것	것	NNB	NNB	_	_	_	_
19	은	은	JX	JX	_	_	_	_

CoNLL-U Format

SyntaxNet Parser

SyntaxNet Parser (1/2)

- Parser CoNLL-U Output

HEAD (parent node ID)

ID	FORM	LEMMA	UPOSTAG	XPOSTAG	FEATS		DEPREL	DEPS
11	회사	_	NNG	NNG	_	12	MOD	_
12	로	_	JKB	JKB	_	13	NP_AJT	_
13	분할	_	NNG	NNG	_	14	MOD	_
14	하	_	XSV	XSV	_	15	MOD	_
15	기	_	ETN	ETN	_	16	MOD	_
16	로	_	JKB	JKB	_	60	VP_AJT	_
17	한	_	MM	MM	_	18	DP	_
18	것	_	NNB	NNB	_	19	MOD	_
19	은	_	JX	JX	_	60	NP_SBJ	_

Modifier

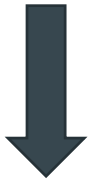
Eojeol Group

Eojeol Reassembly

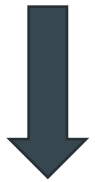
- Component->Eojeol Reassembly
  - Recombine prior MOD components into final Eojeol
  - Remap all tree node IDs

ID	FORM	LEMMA	HEAD	DEPREL
6	회사로	회사/NNG + 로/JKB	7	NP_AJT
7	분할하기로	분할/NNG + 하/XSV + 기/ETN + 로/JKB	27	VP_AJT
8	한	한/MM	9	DP
9	것은	것/NNB + 은/JX	27	NP_SBJ

SyntaxNet Parser



Component->Eojeol Reassembly

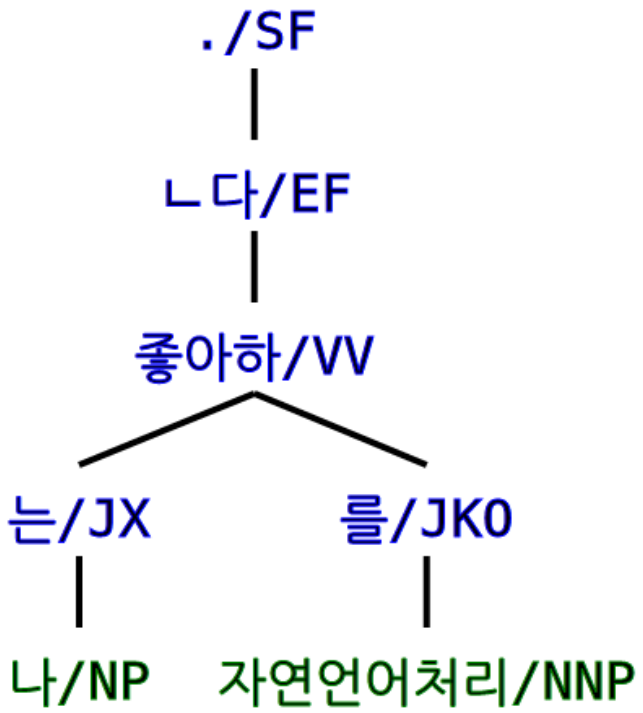


Tree Visualization (SVG)

Tree Visualization (1/3)

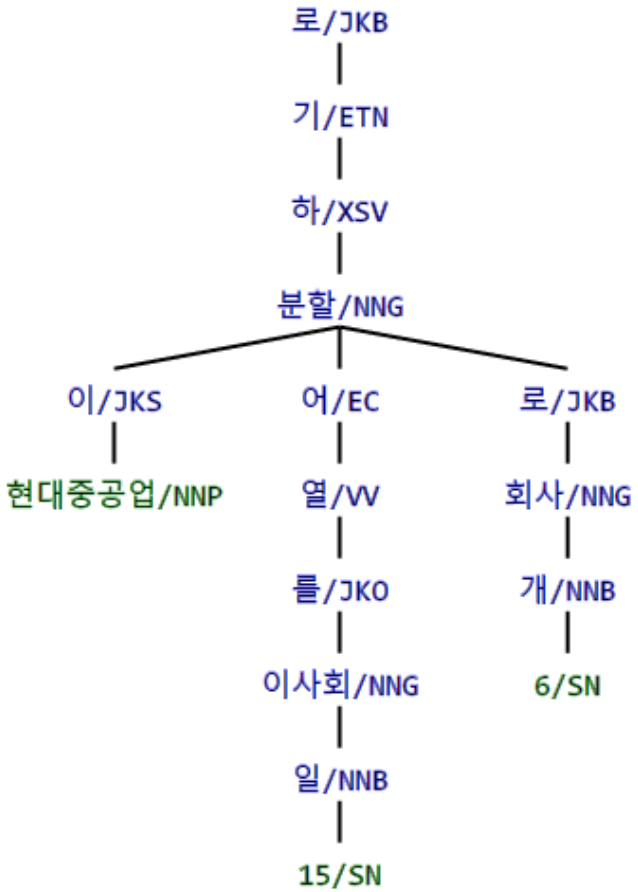
Given CoNLL-U format, we can easily generate a tree.

```
childList = {}
for elem in conllTable:
  if not(elem.HEAD in childList):
    childList[elem.HEAD] = []
  childList[elem.HEAD].append(elem)
  if(elem.HEAD == 0):
    rootElem = elem

def makeNode(elem, childList):
  node = Node()
  if elem.ID in childList:
    for child in childList[elem.ID]:
      node.children.append(makeNode(child, childList))
  return node

rootNode = makeNode(rootElem, childList)
```



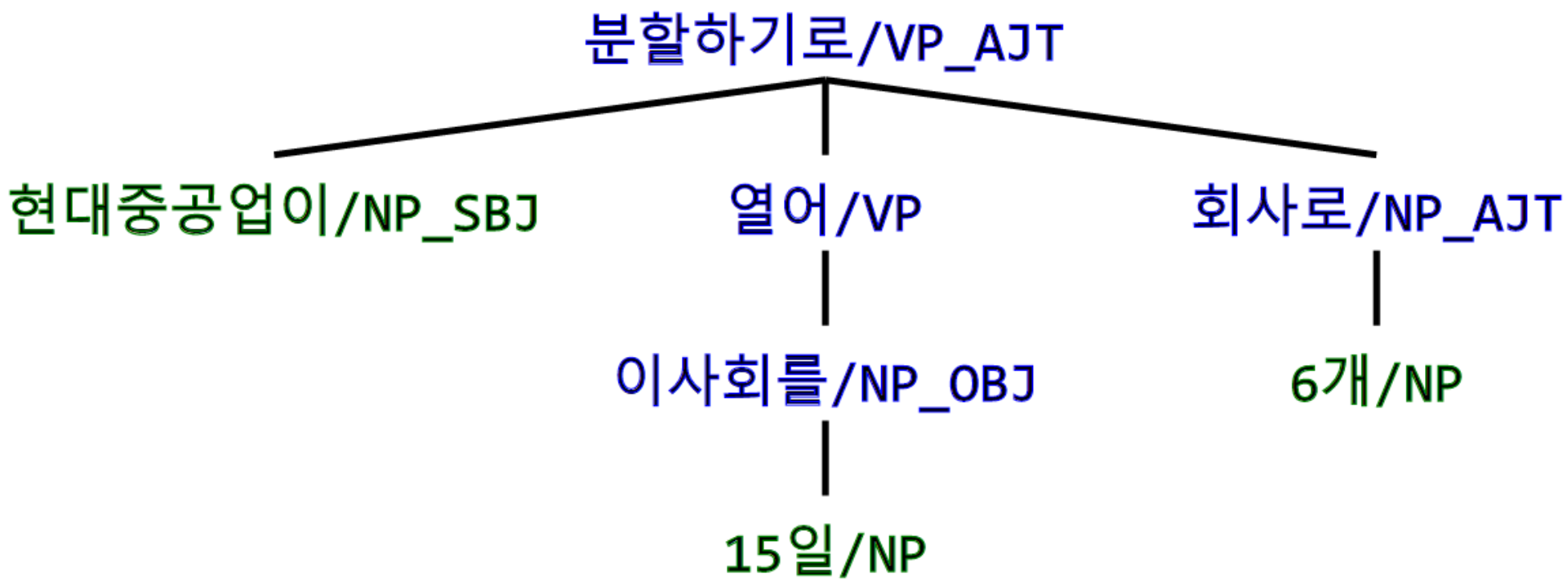
Tree Visualization (2/3)

Componentized Output Tree



Tree Visualization (3/3)

Reassembled Output Tree



Serving Model

- TensorFlow Serving
  - Server-client based API model
  - Requires custom patch from git
  - Still buggy for SyntaxNet
- SyntaxNet demo scripts
  - Pipe from stdin/stdout
  - Messy, but most reliable

Example (1/5)

- Python-hosted script on Amazon EC2 which invokes SyntaxNet
- Generates diagram of dependency parsing tree from CoNLL-U output
- Modified tree generator to output SVG format for flexibility

- http://andrewmatteson.name/psg_tree.htm

Example (2/5)

Sample Sentence

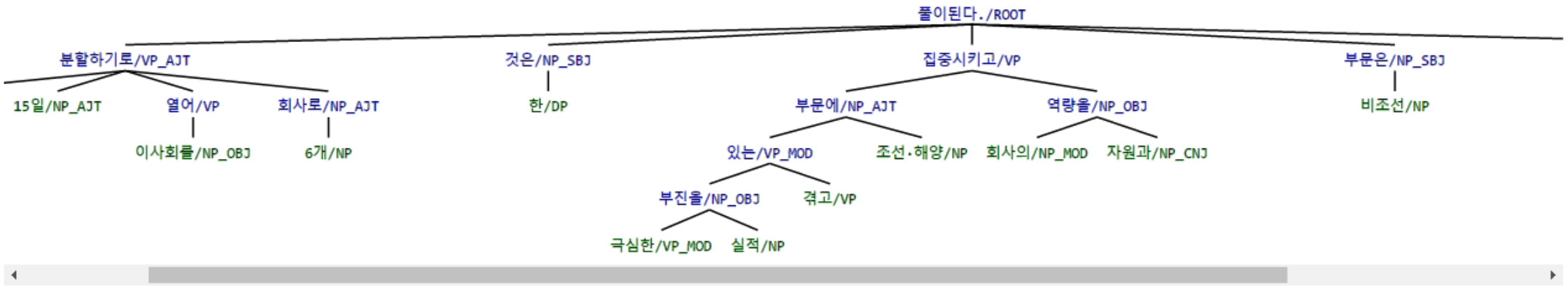
- 현대중공업이 15일 이사회를 열어 6개 회사로 분할하기로 한 것은 극심한 실적 부진을 겪고 있는 조선·해양 부문에 회사의 자원과 역량을 집중시키고 비조선 부문은 자생력을 키워주기 위한 '고육지책'으로 풀이된다.

Example (3/5)

SyntaxNet Tree Generator (Korean/Komoran3)

Sentence to Parse: 현대중공업이 15일 이사회를 열어 6개 회사로 분할하기로 한 것은 극심한 실적 부진을 겪고 있는 조선·해양 부문에 회사의 자원과 역량을 집중시키고 비조선 부문은 자생력을 키워주기 위한 '고육지책'으로 풀이된다. Submit

Visualization



Example (4/5)

CoNLL-U Output

Details

1 POS Tagging | 2 Dependency Parsing | 3 Reassembly | 4 Semantic Analysis | D Debug Logs

ID	FORM	LEMMA	UPOSTAG	XPOSTAG	FEATS	HEAD	DEPREL
1	현대중공업이	현대중공업/NNP + 이/JKS	_	_	_	7	NP_SBJ
2	15일	15/SN + 일/NNB	_	_	_	7	NP_AJT
3	이사회를	이사회/NNG + 를/JKO	_	_	_	4	NP_OBJ
4	열어	열/VV + 어/EC	_	_	_	7	VP
5	6개	6/SN + 개/NNB	_	_	_	6	NP
6	회사로	회사/NNG + 로/JKB	_	_	_	7	NP_AJT
7	분할하기로	분할/NNG + 하/XSV + 기/ETN + 로/JKB	_	_	_	27	VP_AJT
8	한	한/MM	_	_	_	9	DP
9	것은	것/NNB + 은/JX	_	_	_	27	NP_SBJ
10	극심한	극심/XR + 하/XSA + ㄴ/ETM	_	_	_	12	VP_MOD
11	실적	실적/NNG	_	_	_	12	NP
12	부진을	부진/NNG + 을/JKO	_	_	_	14	NP_OBJ

Example (5/5)

1 POS Tagging | 2 Dependency Parsing | 3 Reassembly | 4 Semantic Analysis | D Debug Logs

ID	FORM	DEPREL	Categories	Synonyms
1	현대중공업이	NP_SBJ	1978년 해체 중립성에 이의가 제기된 문서/2013년 5월 대한민국의 조선업 기업 대한민국의 중공업 기업 한국거래소 상장 기업 현대중공업그룹 출처가 필요한 글/2016년 2월 대한민국의 군수산업체 1978년 설립 1973년 설립	Hyundai Heavy Industries 현대조선중공업 현대중공업
2	15일	NP_AJT	동음이의어 문서	
3	이사회를	NP_OBJ	회사법 상법 경영	Board of directors 이사회
4	열어	VP		

감사합니다.

Andrew Matteson
E-mail : amatteson@korea.ac.kr