Context, Syntax and SyntaxNet

2016.09.05

Jaemin Cho

Introduction

Today we will cover..

- Context in NLP
- How to run SyntaxNet

Introduction

Today we will not cover..

- Conventional NLP
- LSTM
- TensorFlow Protobuf
- CRF

Context in NLP

Main Tasks in NLP

- Language Modeling
- Translation
- Question Answering
- Summarization

● Related tasks

- POS (Part-of-Speech) Tagging
- Parsing

Context in NLP

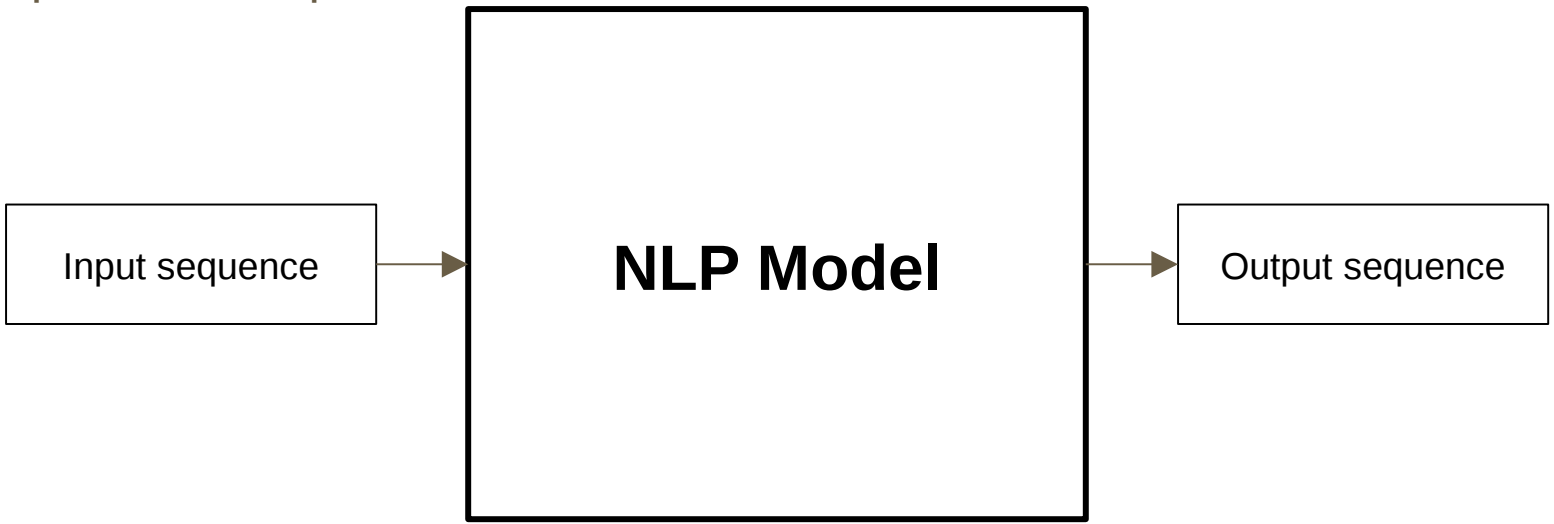
Main Tasks in NLP

- Word Embedding ex) One-hot encoding, Word2Vec
- Assign probability to word ex) N-gram, CBOW, Skip-gram, Word2Vec
- Vocabulary / Out-of-vocab Words
- Tokenization /lemmatization ex) going, goes, went => go

Context in NLP

Main Tasks in NLP

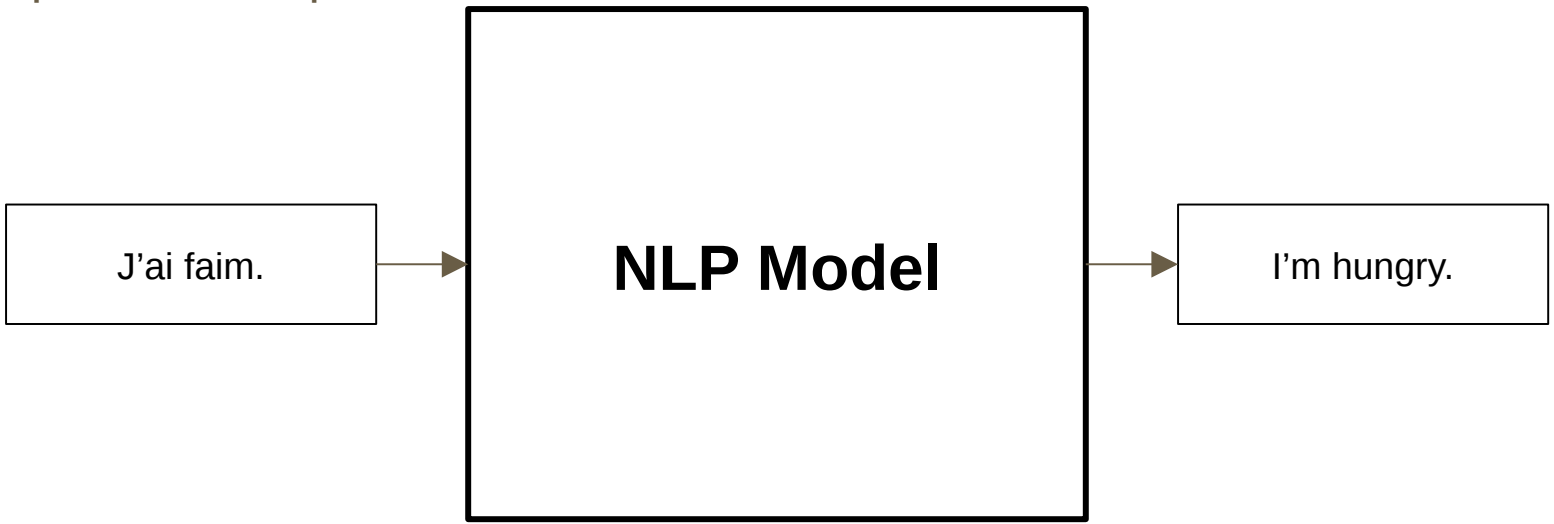
- Input => Output



Context in NLP

Main Tasks in NLP

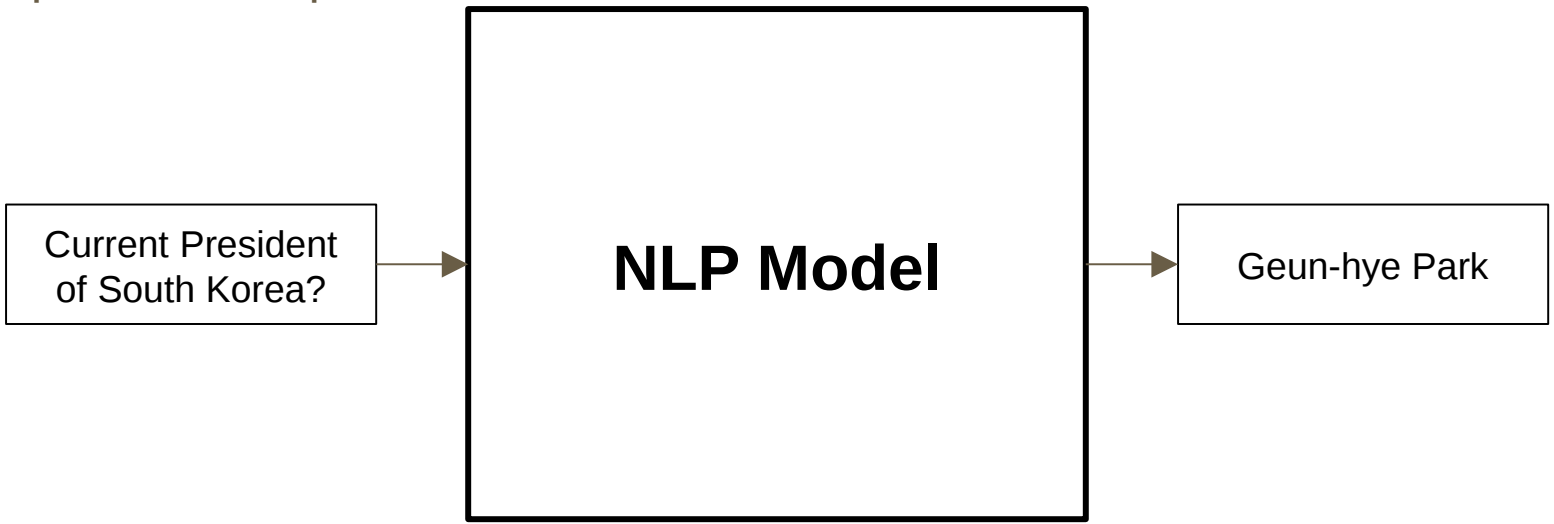
- Input => Output



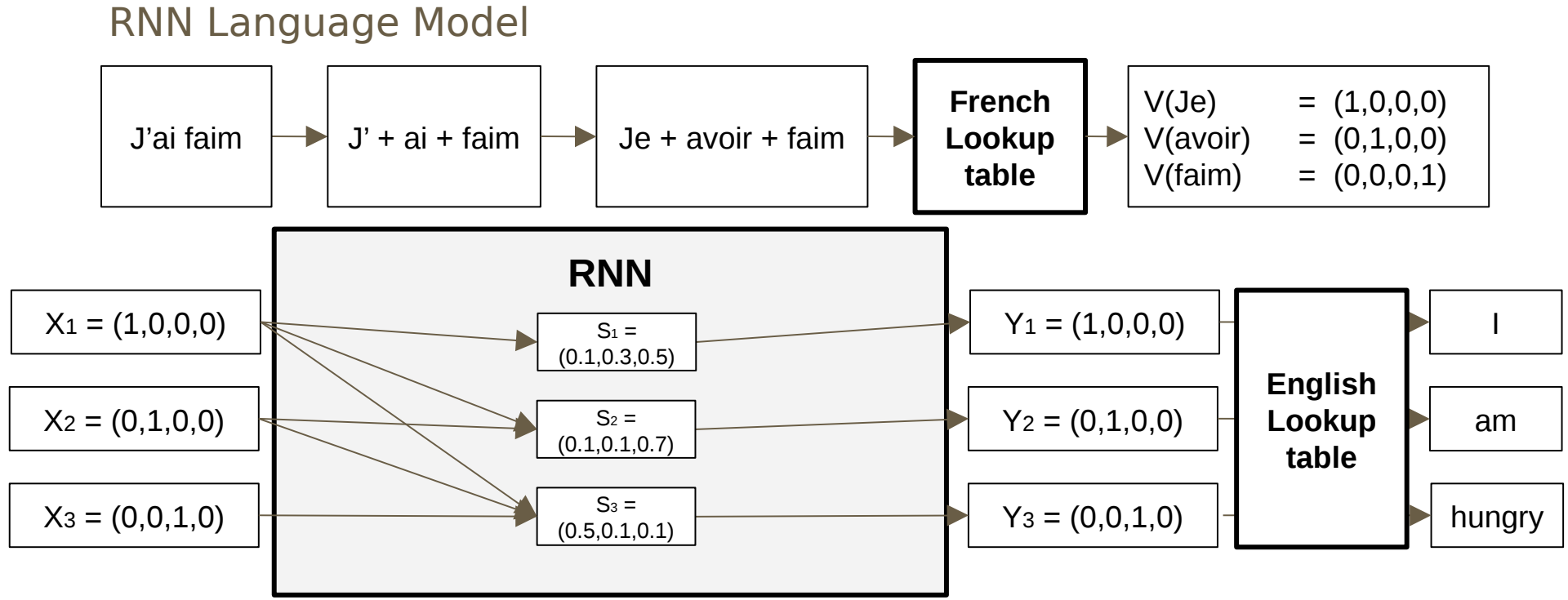
Context in NLP

Main Tasks in NLP

- Input => Output

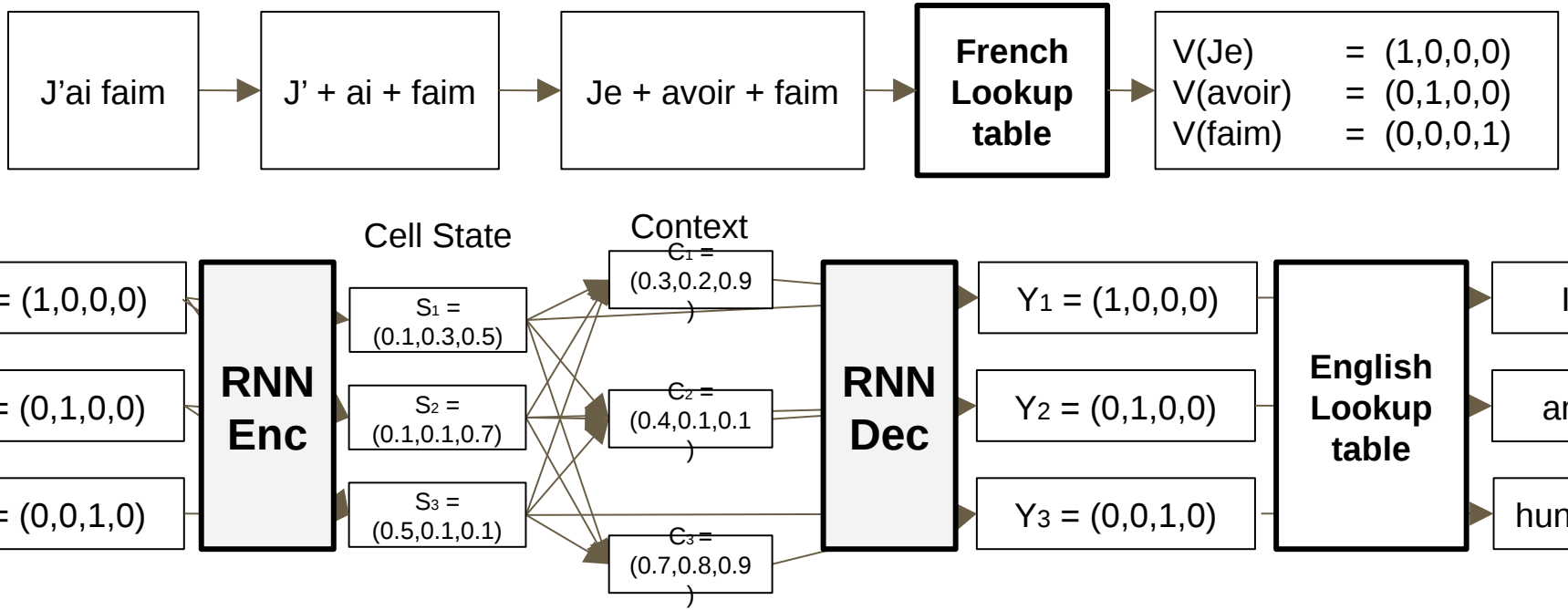


Context in NLP



Context in NLP

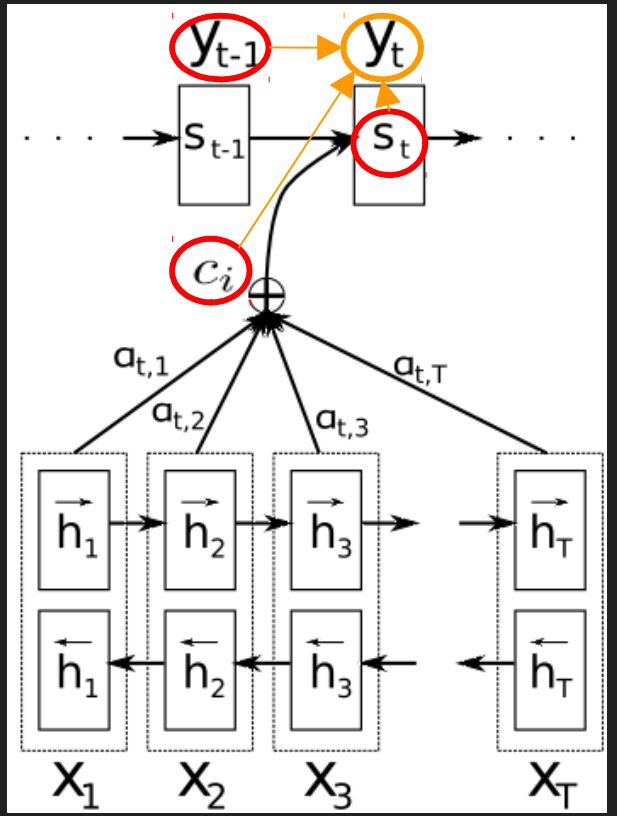
RNN Encoder-Decoder Model + Attention



Basic Attention Model

Softmax

$$p(y_i|s_i, y_{i-1}, c_i) \propto \exp\left(y_i^\top W_o t_i\right)$$



Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473,

Basic Attention Model

Softmax

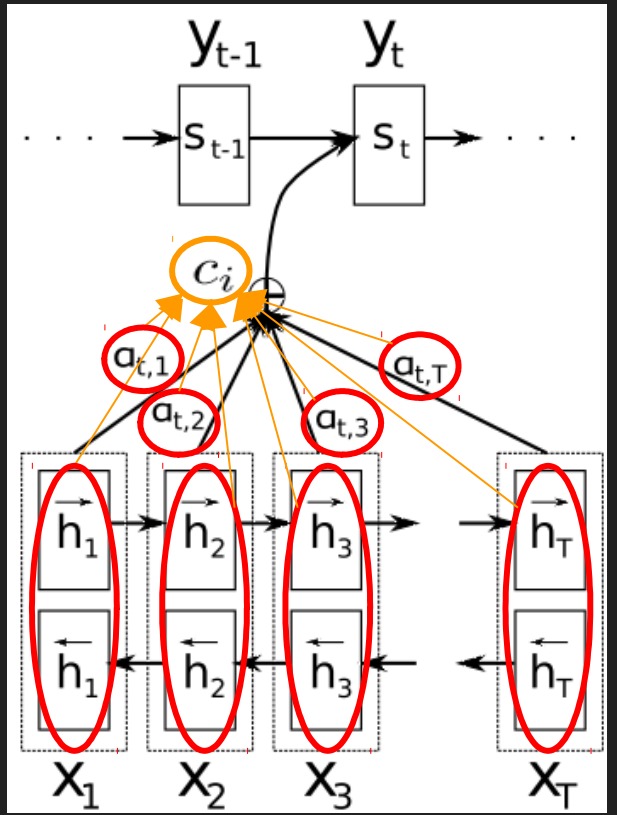
$$p(y_i|s_i, y_{i-1}, c_i) \propto \exp\left(y_i^\top W_o t_i\right)$$

Context

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

Weight of h

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{T_x} \exp\left(e_{ik}\right)}$$



Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473,

Basic Attention Model

Softmax

$$p(y_i|s_i, y_{i-1}, c_i) \propto \exp\left(y_i^\top W_o t_i\right)$$

Context

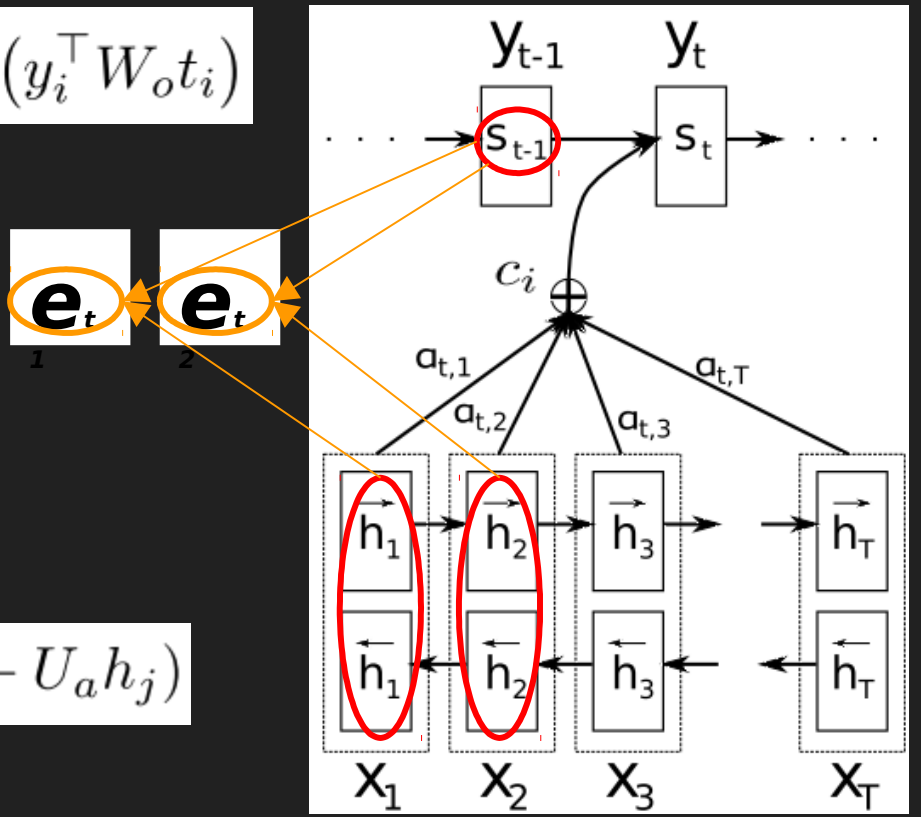
$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

Weight of h

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{T_x} \exp\left(e_{ik}\right)}$$

Annotation

$$e_{ij} = v_a^\top \tanh\left(W_a s_{i-1} + U_a h_j\right)$$



Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473,

Basic Attention Model

Softmax

$$p(y_i|s_i, y_{i-1}, c_i) \propto \exp\left(y_i^\top W_o t_i\right)$$

Context

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

Weight of h

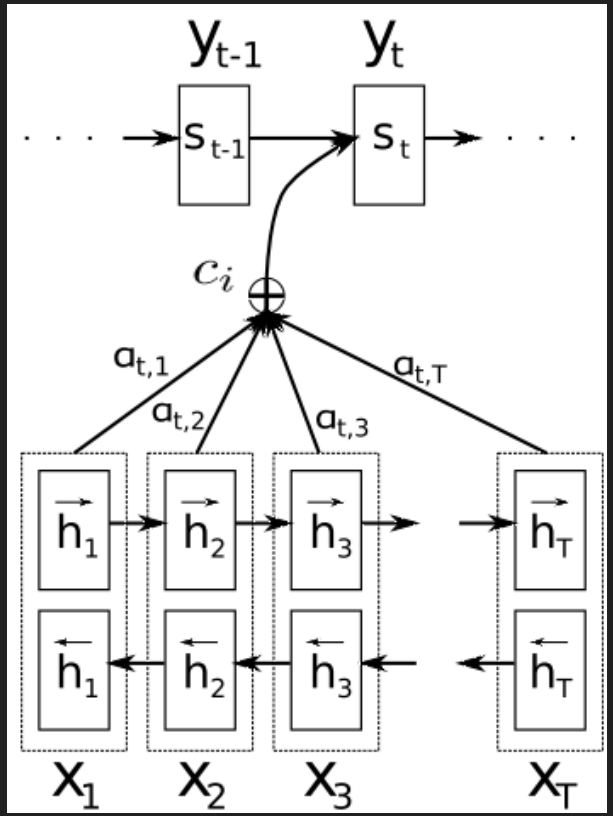
$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{T_x} \exp\left(e_{ik}\right)}$$

Annotation

$$e_{ij} = v_a^\top \tanh\left(W_a s_{i-1} + U_a h_j\right)$$

Maxout

$$t_i = \left[\max\left\{\tilde{t}_{i,2j-1}, \tilde{t}_{i,2j}\right\}\right]_{j=1,\ldots,l}^\top \qquad \tilde{t}_i = U_o s_{i-1} + V_o E y_{i-1} + C_o c_i$$



Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473,

Context in NLP

What is Context?

Context in NLP

context [kɑ:ntekst]

명사

1. (어떤 일의) 맥락, 전후 사정

This speech needs to be set in the context of Britain in the 1960s.

이 연설은 1960년대의 영국이라는 맥락 속에 두고 볼 필요가 있다.

2. (글의) 맥락, 문맥

You should be able to guess the meaning of the word from the context.

문맥을 통해 단어의 의미를 추측할 수 있어야 한다.

NAVER 영어사전, 2016

Context in NLP

“Structure of any discourse is a composite of **three distinct but interacting components”**

Grosz, Barbara J and Sidner, Candace L, “Attention, intentions, and the structure of discourse” (1986)

- the structure of the actual sequence of utterances in the discourse;
- a structure of intentions;
- an attentional state.

Context in NLP

Semantic Context

의미, 맥락
Intention, Attention, Tone

갑자기 딴소리 안하기
인격 유지
말투 유지

Syntactic Context

문법 (Physical Context)
Grammar, Rule

Domain, 언어에 따라 다양한 규칙
ex) German, Python Code

POS Tagging (형태소 분석)
Parsing (구문 분석)

Context in NLP

- 단순 End-to-End RNN Generative Model로는 특정 코퍼스를 모델링하는 데 한계가 있음
  => 유한한 코퍼스의 한계
  => Meaning loss when Embedding words to Vectors
  => 언어 특징에 특화된 모델, 추가적인 규칙이 필요함

  ex) 한글
  "주어 다음에 조사가 와야 한다.", "음운의 끝소리에 따라 호응하는 조사가 다르다"
  ex) 파이썬 코드
  "띄어쓰기", "괄호를 열었으면 괄호를 닫아야 한다"
  ex) 챗봇
  "자아 유지", "주제 유지"

Context in NLP

- 품사 / 문장성분에 따라 다른 디코더 사용, 결과물 조합

  - 현재 입력받은 단어 / 현재 출력할 단어가 어떤 성분인지 파악해야 함
    => Probabilistic Graphical Models (HMM, CRF...)

  - 디코딩한 출력들을 다시 조합해야 함
    => POS Tagging (형태소 분석), Parsing (구문 분석)

Context in NLP

- Until 2015, RNN Enc-Dec models have had **only one segment** that consists of all the utterances => **Attention (context vector)**

- Some recent variations

- Effective Approaches to Attention-based Neural Machine Translation (2015)
  => Global Attention + Local Attention
- Attention with Intention for a Neural Network Conversation Model (2015)
  => Attention Vector + Intention Vector
- Incorporating Copying Mechanism in Sequence-to-Sequence Learning (2016)
  => P(y_t) = P_generate(y_t) + P_copy(y_t)
- Latent Predictor Networks for Code Generation (2016)
  => Semi-Markov CRF + Multiple Decoder (pointer networks) for different input sequences

Context in NLP

Semantic Context

의미, 맥락
Intention, Attention, Tone

갑자기 딴소리 안하기
인격 유지
말투 유지

Syntactic Context

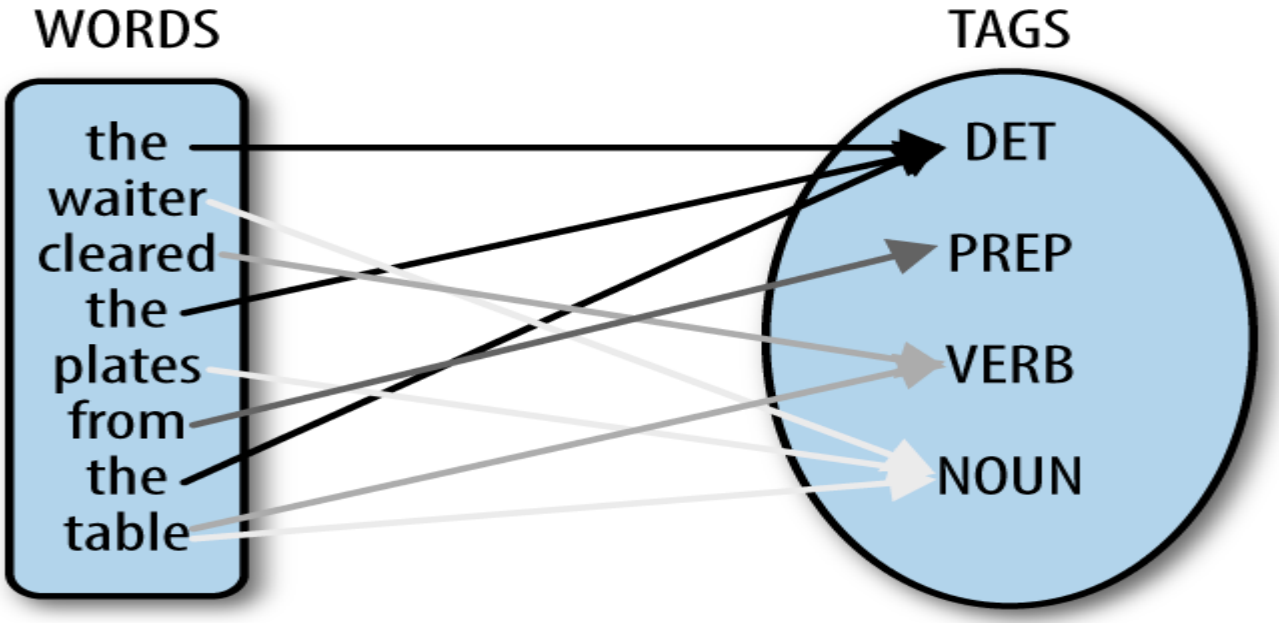
문법 (Physical Context)
Grammar, Rule

Domain, 언어에 따라 다양한 규칙
ex) German, Python Code

POS Tagging (형태소 분석)
Parsing (구문 분석)

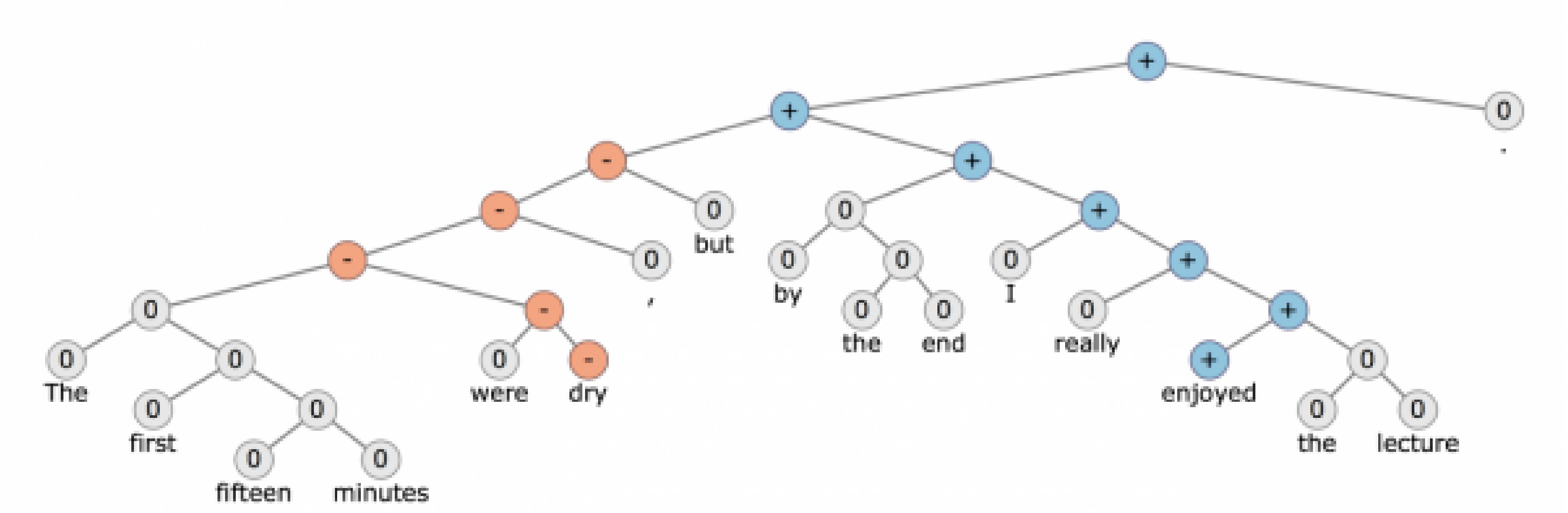
Context in NLP

- POS Tagging



Context in NLP

- Parsing



Context in NLP

- NLP package for Python
  - NLTK (Natural Language Toolkit)
    - 여러 가지 코퍼스, POS Tagger, Parser 외 여러 가지 기능 제공
    - http://www.nltk.org/
    - Sentdex’s NLTK Tutorial https://www.youtube.com/watch?v=FLZvOKSCkxY
  - KoNLPy
    - KoNLP (R 기반) Wrapper
    - Mecab-ko, Hannanum, Kkma, Komoran 형태소 분석기 제공
    - http://konlpy.org/ko/v0.4.4/

Context in NLP

●Korean Corpus

- 고려대학교 말뭉치
- HANTEC 2.0
- HKIB-40075
- KAIST Corpus
- **Sejong Corpus**
- 연세 말뭉치
- BoRA 언어자원은행 제공

SyntaxNet

SyntaxNet: Neural Models of Syntax.

A TensorFlow implementation of the models described in Andor et al. (2016).

Update: Parsey models are now available for 40 languages trained on Universal Dependencies datasets, with support for text segmentation and morphological analysis.

At Google, we spend a lot of time thinking about how computer systems can read and understand human language in order to process it in intelligent ways. We are excited to share the fruits of our research with the broader community by releasing SyntaxNet, an open-source neural network framework for TensorFlow that provides a foundation for Natural Language Understanding (NLU) systems. Our release includes all the code needed to train new SyntaxNet models on your own data, as well as *Parsey McParseface*, an English parser that we have trained for you, and that you can use to analyze English text.

https://github.com/tensorflow/models/tree/master/syntaxnet

SyntaxNet

So, how accurate is Parsey McParseface? For this release, we tried to balance a model that runs fast enough to be useful on a single machine (e.g. ~600 words/second on a modern desktop) and that is also the most accurate parser available. Here's how Parsey McParseface compares to the academic literature on several different English domains: (all numbers are % correct head assignments in the tree, or unlabelled attachment score)

Model	News	Web	Questions
Martins et al. (2013)	93.10	88.23	94.21
Zhang and McDonald (2014)	93.32	88.65	93.37
Weiss et al. (2015)	93.91	89.29	94.17
Andor et al. (2016)*	94.44	90.17	95.40
Parsey McParseface	94.15	89.08	94.77

We see that Parsey McParseface is state-of-the-art; more importantly, with SyntaxNet you can train larger networks with more hidden units and bigger beam sizes if you want to push the accuracy even further: Andor et al. (2016)* is simply a SyntaxNet model with a larger beam and network. For futher information on the datasets, see that paper under the section "Treebank Union".

Parsey McParseface is also state-of-the-art for part-of-speech (POS) tagging (numbers below are per-token accuracy):

Model	News	Web	Questions
Ling et al. (2015)	97.78	94.03	96.18
Andor et al. (2016)*	97.77	94.80	96.86
Parsey McParseface	97.52	94.24	96.45

https://github.com/tensorflow/models/tree/master/syntaxnet

SyntaxNet

- Transition Based (vs Graph Based) Dependency Parsing
- Pretrain (Greedy) + Train (CRF)

- “Globally Normalized Transition-Based Neural Networks” Andor et al. (2016)
- http://arxiv.org/abs/1603.06042

SyntaxNet

Graph-based Dependency Parsing

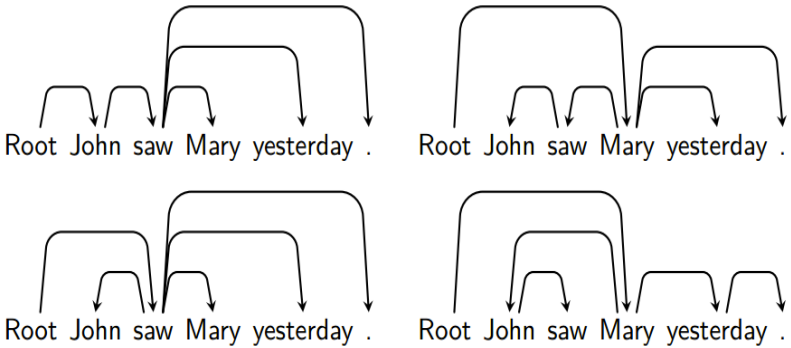
- Find the highest scoring tree from a complete dependency graph.

$$Y^* = \arg\max_{Y \in \Phi(X)} score(X, Y)$$

http://images.slideplayer.com/23/6623811/slides/slide_14.jpg

Graph-based dependency parsing

Candidate trees for "John saw mary yesterday"



...

http://www.coli.uni-saarland.de/~yzhang/rapt-ws1112/slides/qiu.pdf

SyntaxNet

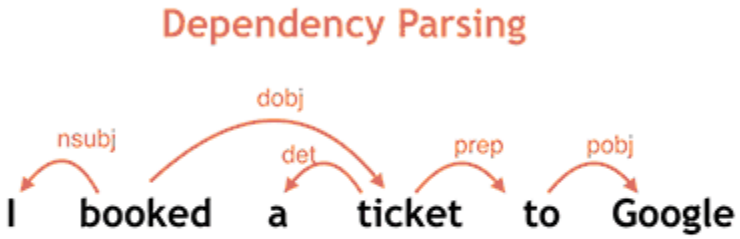
- Transition Based Dependency Parsing

Given an input x, most often a sentence, we define:

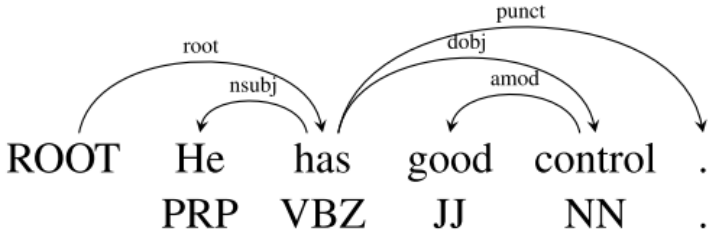
- A set of states $\mathcal{S}(x)$.
- A special start state $s^{\dagger} \in \mathcal{S}(x)$.
- A set of allowed decisions $\mathcal{A}(s, x)$ for all $s \in \mathcal{S}(x)$.
- A transition function $t(s, d, x)$ returning a new state s' for any decision $d \in \mathcal{A}(s, x)$.

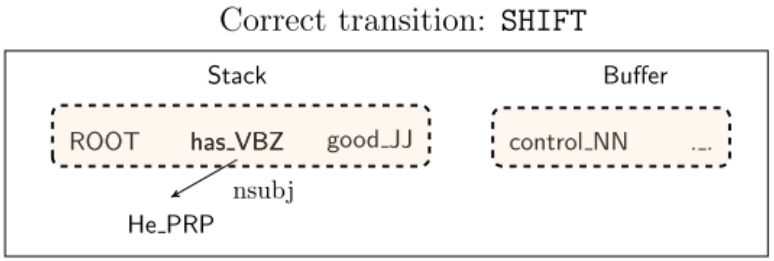
SyntaxNet

- Transition Based Dependency Parsing



SyntaxNet





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 and Manning, "A Fast and Accurate Dependency Parser using Neural Networks" (2014)

SyntaxNet

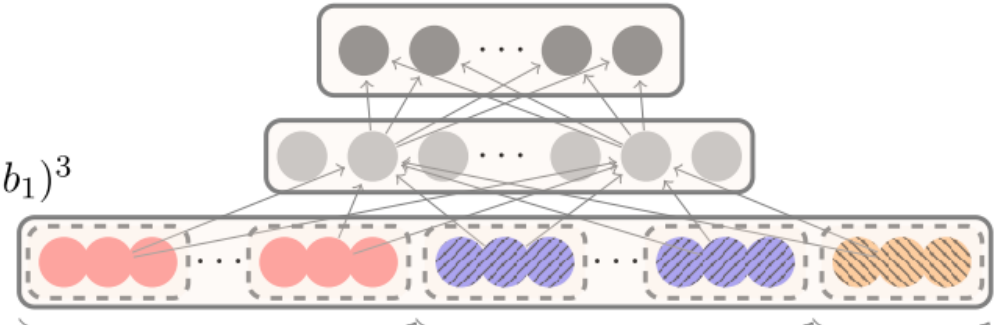
Softmax layer:

$$p = \texttt{softmax}(W_2 h)$$

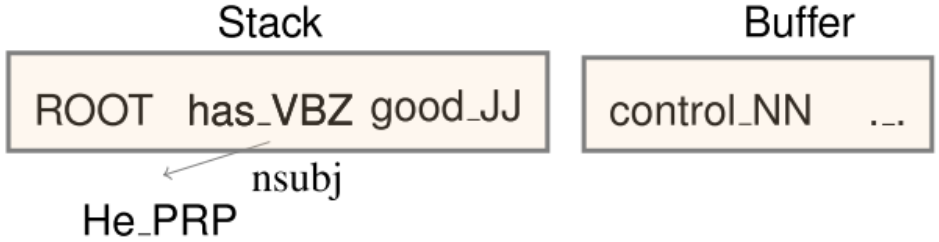
Hidden layer:

$$h = (W_1^w x^w + W_1^t x^t + W_1^l x^l + b_1)^3$$

Input layer: $[x^w, x^t, x^l]$



Configuration



Chen and Manning, "A Fast and Accurate Dependency Parser using Neural Networks" (2014)

SyntaxNet

- Local Pretrain (Greedy) + Global Train (CRF)

$$\rho(s, d; \theta) = \phi(s; \theta^{(l)}) \cdot \theta^{(d)}$$

$$p_L(d_{1:n}) = \prod_{j=1}^{n} p(d_j | d_{1:j-1}; \theta) = \frac{\exp \sum_{j=1}^{n} \rho(d_{1:j-1}, d_j; \theta)}{\prod_{j=1}^{n} Z_L(d_{1:j-1}; \theta)}.$$

$$Z_L(d_{1:j-1}; \theta) = \sum_{d' \in \mathcal{A}(d_{1:j-1})} \exp \rho(d_{1:j-1}, d'; \theta).$$

$$p_G(d_{1:n}) = \frac{\exp \sum_{j=1}^{n} \rho(d_{1:j-1}, d_j; \theta)}{Z_G(\theta)},$$

where

$$Z_G(\theta) = \sum_{d'_{1:n} \in \mathcal{D}_n} \exp \sum_{j=1}^{n} \rho(d'_{1:j-1}, d'_j; \theta)$$

SyntaxNet

- How to Use SyntaxNet
  - Need Corpora with CoNLL-U format

```
1-2   He's        _          _        _        _                                  _   _        _   _
1     He          he         PRON     PRP      Case=Nom|Number=Sing|Person=3      2   nsubj    _   _
2     is          be         VERB     VBZ      Number=Sing|Person=3|Tense=Pres    0   root     _   _
3     in          in         ADP      IN       _                                  6   case     _   _
4     the         the        DET      DT       Definite=Def|PronType=Art          6   det      _   _
5     United      unite      VERB     VBD      Tense=Past|VerbForm=Part           6   amod     _   _
6     Kingdom     kingdom    NOUN     NN       Number=Sing                        2   nmod     _   _
7     (           (          PUNCT    -LRB-    _                                  8   punct    _   SpaceAfter=No
8     UK          UK         PROPN    NNP      Number=Sing                        6   appos    _   SpaceAfter=No
9     )           )          PUNCT    -RRB-    _                                  8   punct    _   SpaceAfter=No
10    .           .          PUNCT    .        _                                  2   punct    _   _
```

SyntaxNet

Live Demo

SyntaxNet

- How to Use SyntaxNet
  - Bazel Build
  - **Docker image**
    - http://blog.nacyot.com/articles/2014-01-27-easy-deploy-with-docker/
    - https://hub.docker.com/r/brianlow/syntaxnet/
- Edit Protobuf (context.pbtxt)
- Edited shell files

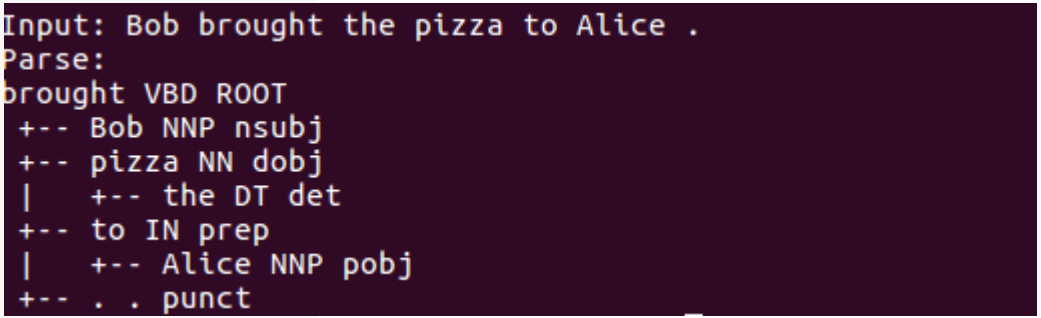
SyntaxNet

docker pull brianlow/syntaxnet => (old version !!)

docker run --name demo -i -t brianlow/syntaxnet bash

```
echo 'Bob brought the pizza to Alice.' | syntaxnet/demo.sh

Input: Bob brought the pizza to Alice .
Parse:
brought VBD ROOT
 +-- Bob NNP nsubj
 +-- pizza NN dobj
 |   +-- the DT det
 +-- to IN prep
 |   +-- Alice NNP pobj
 +-- . . punct
```



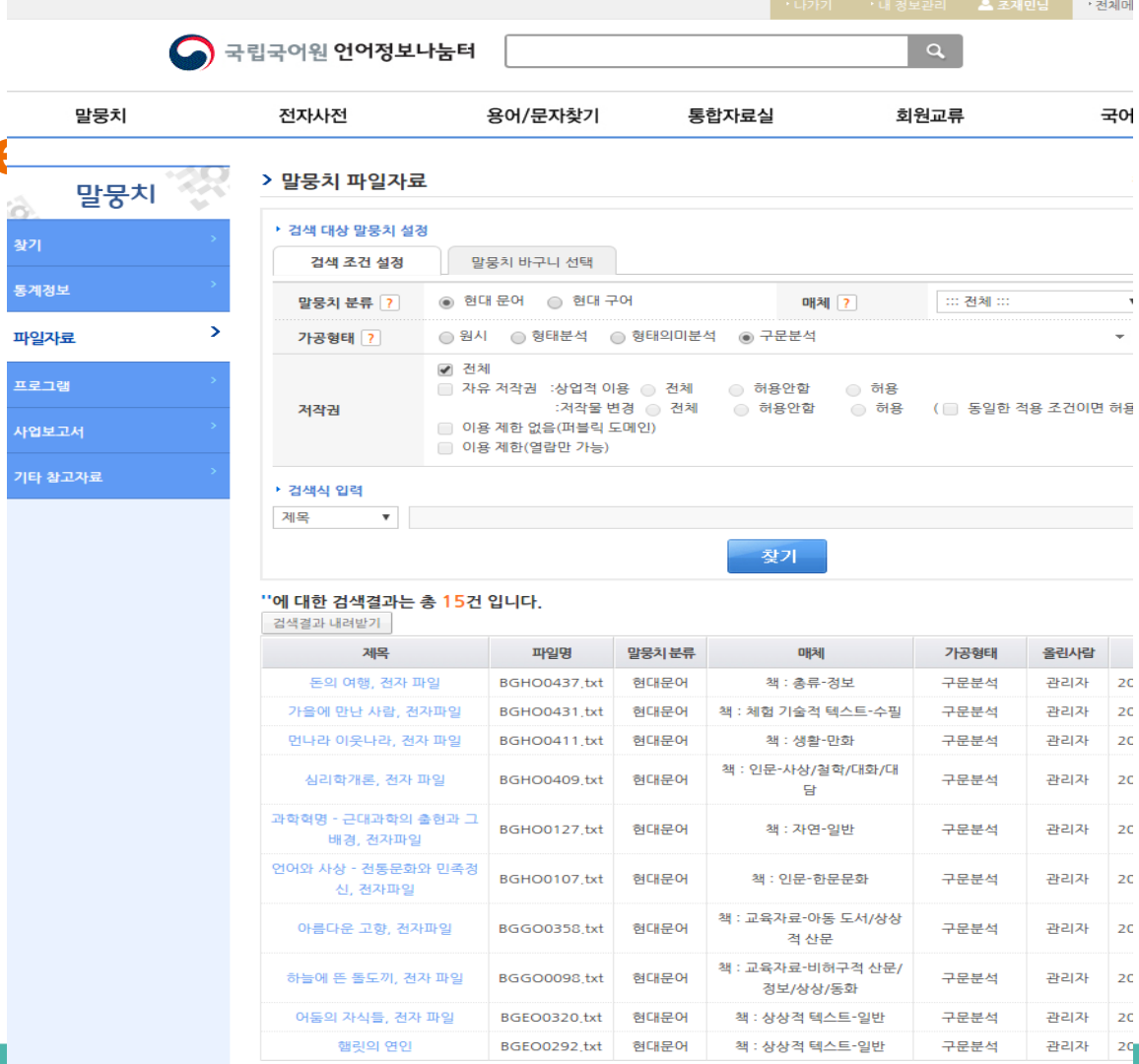
(optional) git clone https://github.com/dasindex/syntaxnet

SyntaxNet

apt-get update

apt-get install language-pack-ko language-pack-ko-base

SyntaxNe


국립국어원 언어정보나눔터
말뭉치
전자사전
용어/문자찾기
통합자료실
회원교류
말뭉치
찾기
통계정보
파일자료
프로그램
사업보고서
기타 참고자료
> 말뭉치 파일자료
검색 대상 말뭉치 설정
검색 조건 설정
말뭉치 바구니 선택
말뭉치 분류
현대 문어
현대 구어
매체
::: 전체 :::
가공형태
원시
형태분석
형태의미분석
구문분석
저작권
전체
자유 저작권 :상업적 이용 전체 허용안함 허용
:저작물 변경 전체 허용안함 허용
이용 제한 없음(퍼블릭 도메인)
이용 제한(열람만 가능)
검색식 입력
제목
찾기
''에 대한 검색결과는 총 15건 입니다.
검색결과 내려받기
제목
파일명
말뭉치 분류
매체
가공형태
올린사람
돈의 여행, 전자 파일
BGHO0437.txt
현대문어
책 : 총류-정보
구문분석
관리자
가을에 만난 사람, 전자파일
BGHO0431.txt
현대문어
책 : 체험 기술적 텍스트-수필
구문분석
관리자
먼나라 이웃나라, 전자 파일
BGHO0411.txt
현대문어
책 : 생활-만화
구문분석
관리자
심리학개론, 전자 파일
BGHO0409.txt
현대문어
책 : 인문-사상/철학/대화/대담
구문분석
관리자
과학혁명 - 근대과학의 출현과 그 배경, 전자파일
BGHO0127.txt
현대문어
책 : 자연-일반
구문분석
관리자
언어와 사상 - 전통문화와 민족정신, 전자파일
BGHO0107.txt
현대문어
책 : 인문-한문문화
구문분석
관리자
아름다운 고향, 전자파일
BGGO0358.txt
현대문어
책 : 교육자료-아동 도서/상상적 산문
구문분석
관리자
하늘에 뜬 돌도끼, 전자 파일
BGGO0098.txt
현대문어
책 : 교육자료-비허구적 산문/정보/상상/동화
구문분석
관리자
어둠의 자식들, 전자 파일
BGEO0320.txt
현대문어
책 : 상상적 텍스트-일반
구문분석
관리자
햄릿의 연인
BGEO0292.txt
현대문어
책 : 상상적 텍스트-일반
구문분석
관리자

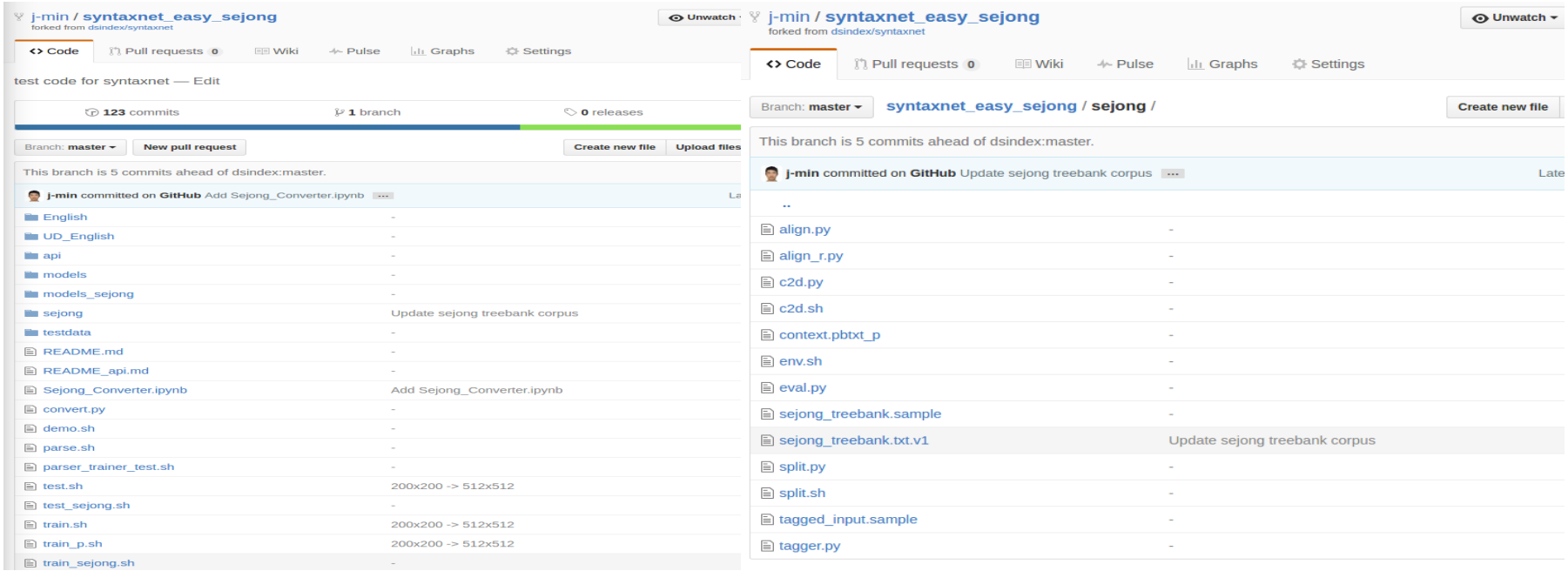
SyntaxNet

총 15개의
현대문어
구문분석 파일

32만 줄
26000 문장

SyntaxNet

git clone https://github.com/j-min/syntaxnet_easy_sejong/



SyntaxNet

git clone https://github.com/j-min/syntaxnet_easy_sejong/

```
$ cd sejong
$ ./split.sh -v -v
$ ls wdir/sejong_treebank.txt.v1.*
wdir/sejong_treebank.txt.v1.test  wdir/sejong_treebank.txt.v1.training
wdir/sejong_treebank.txt.v1.tuning
$ ./c2d.sh -v -v
$ ls wdir/deptree.txt.v3*
wdir/deptree.txt.v3.test  wdir/deptree.txt.v3.training
wdir/deptree.txt.v3.tuning
```

https://github.com/dsindex/syntaxnet/issues/4

SyntaxNet

cd wdir

ls -l

```
root@b155f826497b:~/models/syntaxnet/work/sejong/wdir# ls -l
total 85140
-rw-r--r-- 1 root root  1525444 Sep  5 05:40 deptree.txt.v2.test
-rw-r--r-- 1 root root        0 Sep  5 05:40 deptree.txt.v2.test.err
-rw-r--r-- 1 root root 12287503 Sep  5 05:40 deptree.txt.v2.training
-rw-r--r-- 1 root root        0 Sep  5 05:40 deptree.txt.v2.training.err
-rw-r--r-- 1 root root  1552280 Sep  5 05:40 deptree.txt.v2.tuning
-rw-r--r-- 1 root root        0 Sep  5 05:40 deptree.txt.v2.tuning.err
-rw-r--r-- 1 root root  2707576 Sep  5 05:40 deptree.txt.v3.test
-rw-r--r-- 1 root root 21821804 Sep  5 05:40 deptree.txt.v3.training
-rw-r--r-- 1 root root  2758416 Sep  5 05:40 deptree.txt.v3.tuning
-rw-r--r-- 1 root root  2233111 Sep  5 05:39 sejong_treebank.txt.v1.test
-rw-r--r-- 1 root root 18064630 Sep  5 05:39 sejong_treebank.txt.v1.training
-rw-r--r-- 1 root root  2262275 Sep  5 05:39 sejong_treebank.txt.v1.tuning
-rw-r--r-- 1 root root  1790159 Sep  5 05:40 sejong_treebank.txt.v2.test
-rw-r--r-- 1 root root   383227 Sep  5 05:40 sejong_treebank.txt.v2.test.err
-rw-r--r-- 1 root root 14412619 Sep  5 05:40 sejong_treebank.txt.v2.training
-rw-r--r-- 1 root root  3150240 Sep  5 05:40 sejong_treebank.txt.v2.training.err
-rw-r--r-- 1 root root  1821129 Sep  5 05:40 sejong_treebank.txt.v2.tuning
-rw-r--r-- 1 root root   379252 Sep  5 05:40 sejong_treebank.txt.v2.tuning.err
```

https://github.com/dsindex/syntaxnet/issues/4

SyntaxNet

cd ..

./train_sejong.sh -v -v

echo “이것 파싱해 주세요" | ./test_sejong.sh

https://github.com/dsindex/syntaxnet/issues/4

SyntaxNet

- Training (CPU i3 Laptop)
  - Universal Dependencies_English

    => 약 11시간
  - Sejong Corpus

Thank you