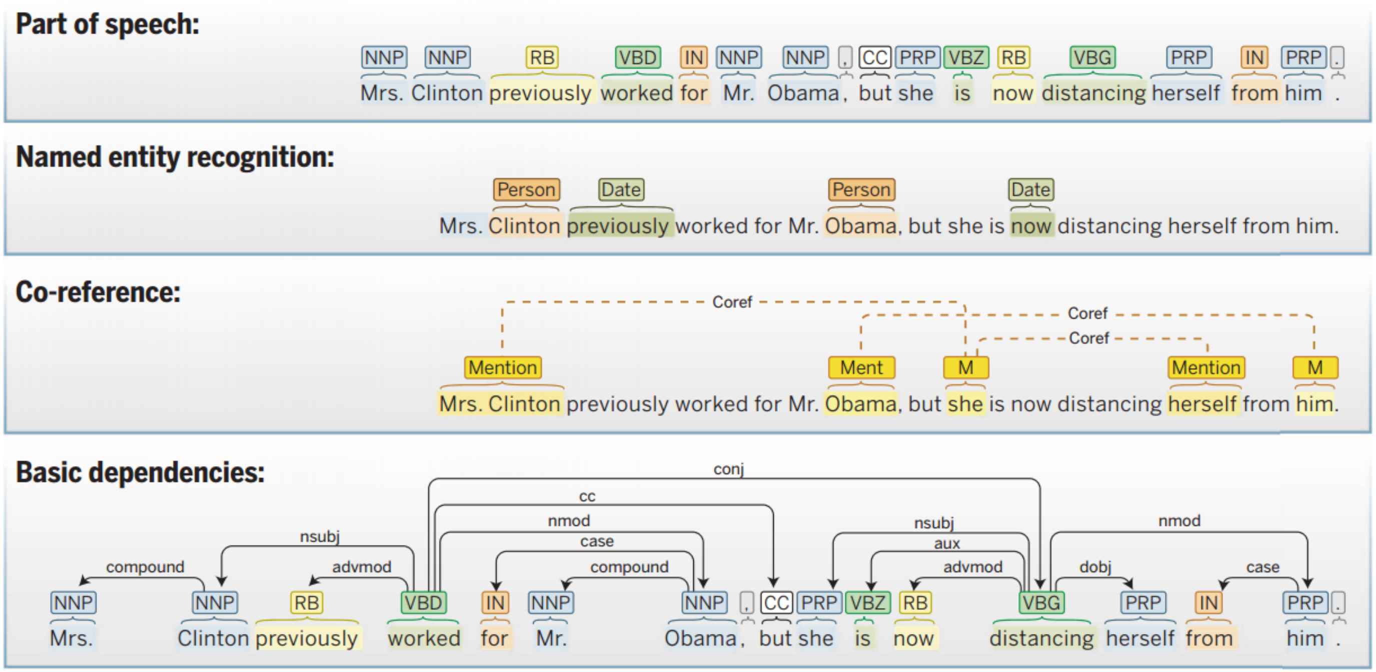


Lecture 2: Text Preprocessing

Pilsung Kang

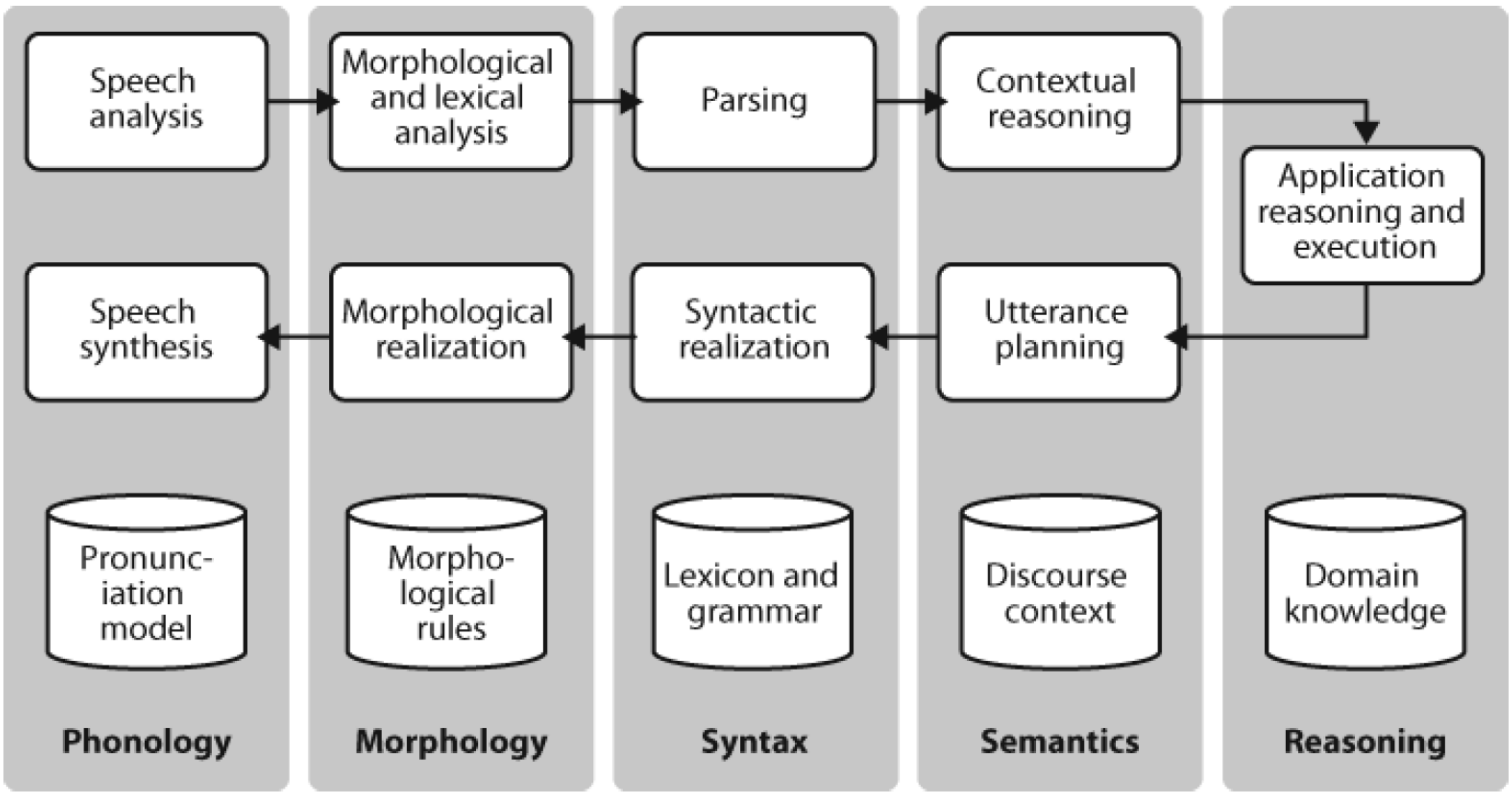
School of Industrial Management Engineering

Korea University

AGENDA

Natural Language Processing

- Natural language processing sequence



Natural Language Processing

Witte (2006)

- Classical categorization of NLP

Classical Categorization

To deal with the complexity of natural langauge, it is typically regarded on several levels (cf. Jurafsky & Martin):

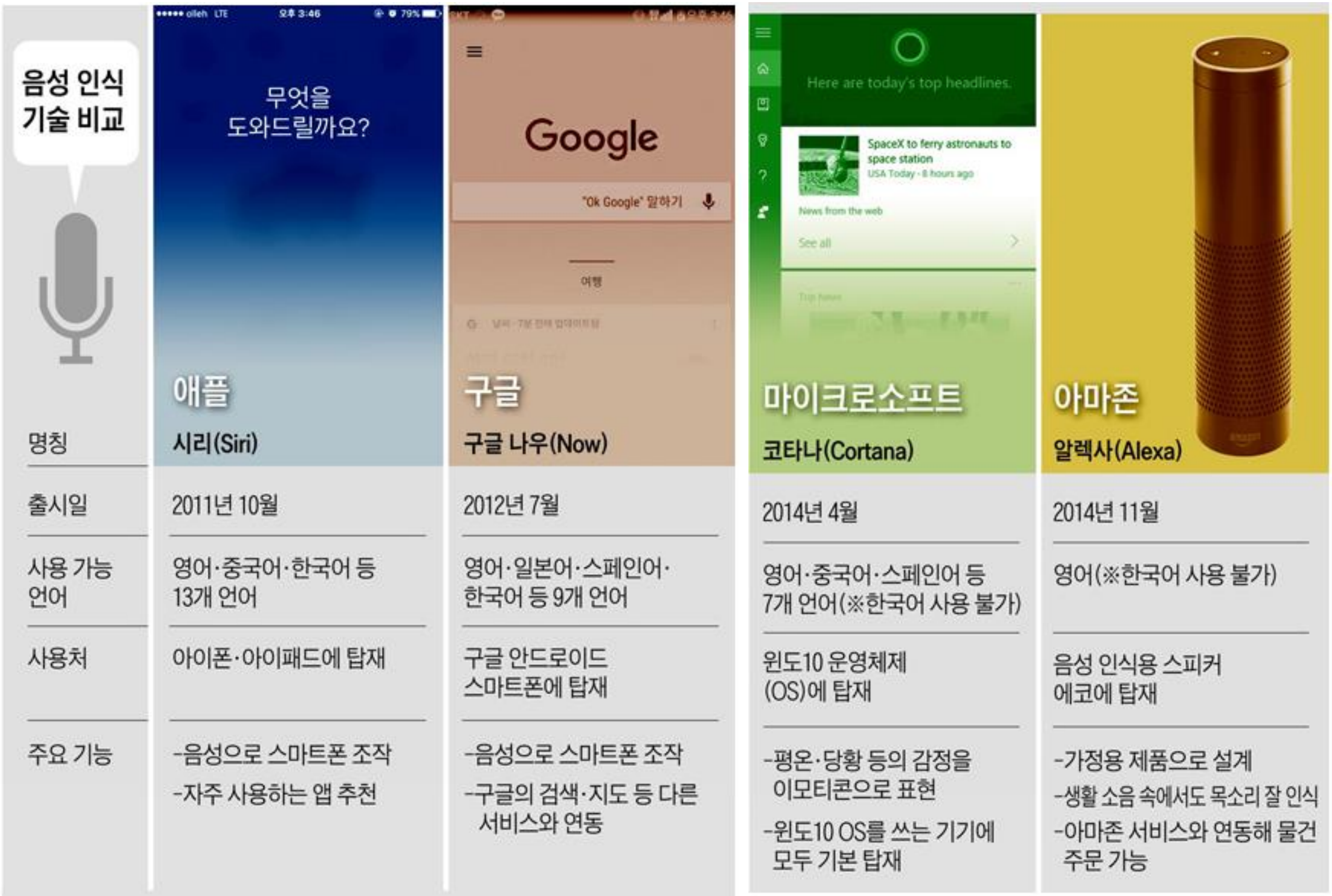
Phonology	the study of linguistic sounds
Morphology	the study of meaningful components of words
Syntax	the study of structural relationships between words
Semantics	the study of meaning
Pragmatics	the study of how language is used to accomplish goals
Discourse	the study of larger linguistic units

Importance for Text Mining

- *Phonology* only concerns spoken language
- *Discourse*, *Pragmatics*, and even *Semantics* is still rarely used

Natural Language Processing

- Phonology is the first gate of AI solutions



음성 인식 기술 비교	애플	구글	마이크로소프트	아마존
명칭	시리(Siri)	구글 나우(Now)	코타나(Cortana)	알렉사(Alexa)
출시일	2011년 10월	2012년 7월	2014년 4월	2014년 11월
사용 가능 언어	영어·중국어·한국어 등 13개 언어	영어·일본어·스페인어·한국어 등 9개 언어	영어·중국어·스페인어 등 7개 언어(※한국어 사용 불가)	영어(※한국어 사용 불가)
사용처	아이폰·아이패드에 탑재	구글 안드로이드 스마트폰에 탑재	윈도10 운영체제(OS)에 탑재	음성 인식용 스피커 에코에 탑재
주요 기능	-음성으로 스마트폰 조작 -자주 사용하는 앱 추천	-음성으로 스마트폰 조작 -구글의 검색·지도 등 다른 서비스와 연동	-평온·당황 등의 감정을 이모티콘으로 표현 -윈도10 OS를 쓰는 기기에 모두 기본 탑재	-가정용 제품으로 설계 -생활 소음 속에서도 목소리 잘 인식 -아마존 서비스와 연동해 물건 주문 가능

http://biz.chosun.com/site/data/html_dir/2016/06/21/2016062100053.html

Natural Language Processing

- Speech to Text (STT)

build passing Kaldi Speech Recognition Toolkit

To build the toolkit: see `./INSTALL`. These instructions are valid for UNIX systems including various flavors of Linux; Darwin; and Cygwin (has not been tested on more "exotic" varieties of UNIX). For Windows installation instructions (excluding Cygwin), see `windows/INSTALL`.

To run the example system builds, see `egs/README.txt`

If you encounter problems (and you probably will), please do not hesitate to contact the developers (see below). In addition to specific questions, please let us know if there are specific aspects of the project that you feel could be improved, that you find confusing, etc., and which missing features you most wish it had.

Kaldi information channels

For HOT news about Kaldi see the project site.

Documentation of Kaldi:

- Info about the project, description of techniques, tutorial for C++ coding.
- Doxygen reference of the C++ code.

Kaldi forums and mailing lists:

We have two different lists

- User list kaldi-help
- Developer list kaldi-developers:

To sign up to any of those mailing lists, go to http://kaldi-asr.org/forums.html:

https://github.com/kaldi-asr/kaldi

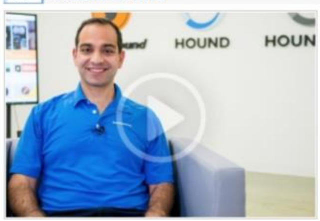
Natural Language Processing

- Top 16 Speech Recognition Startups (2020. 02. 06)

1 Mobvoi

Country: **China** | Funding: **$252.7M**
Mobvoi is an AI company that developed Chinese voice recognition, natural language processing, and vertical search technology in-house.

2 SoundHound



Country: **USA** | Funding: **$115M**
SoundHound develops voice-enabled AI and conversational intelligence technologies. It provides Speech-to-Meaning engine and Deep Meaning Understanding technology that can be built in other services and devices. It also develops app for music recognition and voice assistant for search.

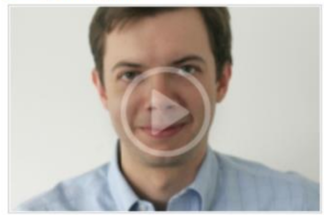
3 Liulishuo

Country: **China** | Funding: **$100M**
Through cutting-edge AI technology & innovative product design, Liulishuo helps users learn English more efficiently and communicate with the world.

4 Invoca

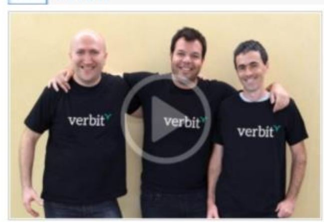
Country: **USA** | Funding: **$60.8M**
Invoca provides complete call intelligence for business. Its machine learning algorithms analyze live phone conversations to understand caller intent and outcomes. Marketers can utilize these insights to make smarter decisions on everything from PPC bidding strategy to digital retargeting audiences.

5 Dialogflow



Country: **USA** | Funding: **$8.6M**
Dialogflow is a conversational user experience platform enabling natural language interactions for devices, applications and services. Developers can use Dialogflow services for speech recognition, natural language processing (intent recognition and context awareness), and conversation management to quickly and easily differentiate their business, increase customer satisfaction and improve business processes.

6 Verbit

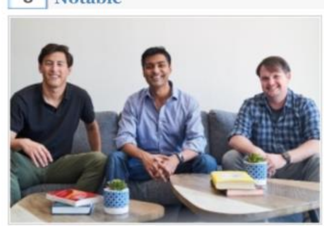


Country: **Israel** | Funding: **$34M**
Verbit is using smart AI technology to disrupt transcription and captioning with automation and speed.

7 Speechmatics

Country: **UK**
Speechmatics provides automatic speech recognition technologies that can be used anywhere, by anyone, in any language.

8 Notable



Country: **USA** | Funding: **$19.2M**
Notable uses AI to automate and digitize every physician-patient interaction. It automtes recording of doctor's visits and updating of electronic health records. The company has developed a technology that uses natural language processing and voice recognition to automatically record doctor-patient interactions and structure the data for inclusion in a patient's medical records.

https://www.ai-startups.org/top/speech_recognition/

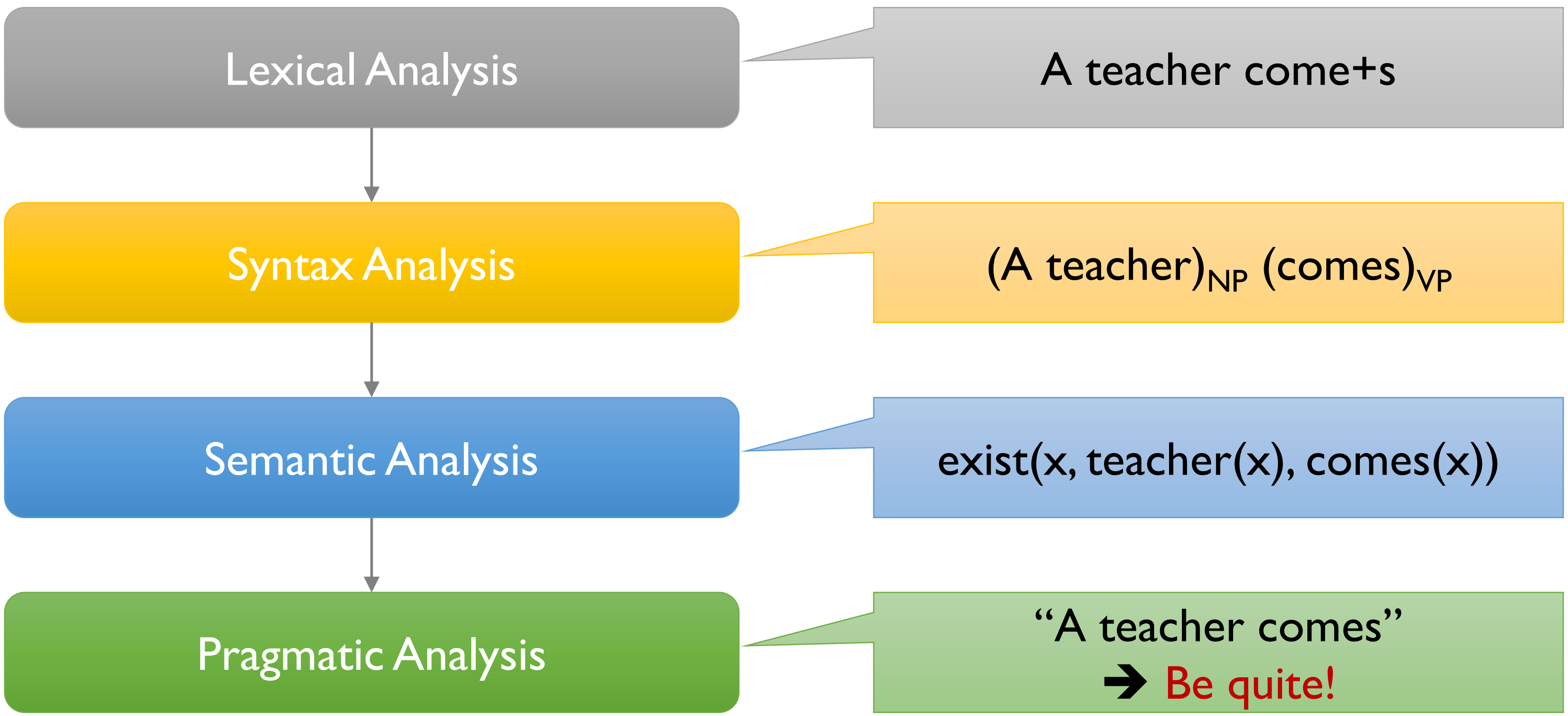
Natural Language Processing

- Text to Speech (TTS) Example

Natural Language Processing

Witte (2006)

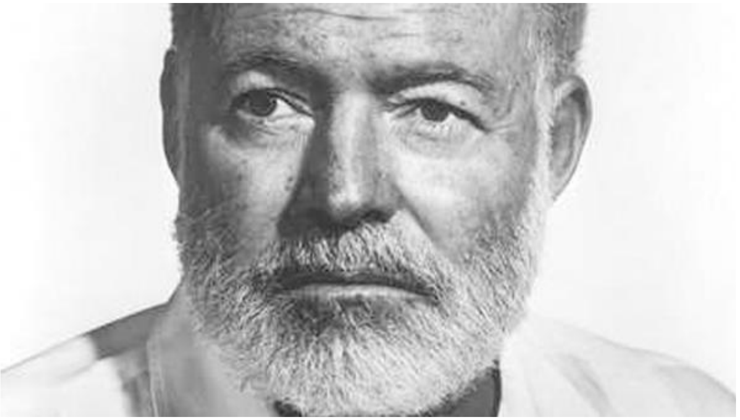
- An example of NLP



Natural Language Processing

- Is Pragmatic Analysis Possible?

“여섯 단어로 우리를 울릴만한
소설을 써 보시지?”



Is NLP Easy? No!

Witte (2006)

- Why is NLP hard?

Difference to other areas in Computer Science

Computer scientist are used to dealing with precise, closed, artificial structures

- e.g., we build a "mini-world" for a database rather than attempting to model every aspect of the real world
- programming languages have a simple syntax (around 100 words) and a precise semantic

This approach does not work for natural language:

- tens of thousands of languages, with more than 100 000 words each
- complex syntax, many ambiguities, constantly changing and evolving

A corollary is that a TM system will never get it "100% right"

Is NLP Easy? No!

- Programming Language

```python
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

import re
import tensorflow as tf


def create_optimizer(loss, init_lr, num_train_steps, num_warmup_steps, use_tpu):
  """Creates an optimizer training op."""
  global_step = tf.train.get_or_create_global_step()

  learning_rate = tf.constant(value=init_lr, shape=[], dtype=tf.float32)

  # Implements linear decay of the learning rate.
  learning_rate = tf.train.polynomial_decay(
      learning_rate,
      global_step,
      num_train_steps,
      end_learning_rate=0.0,
      power=1.0,
      cycle=False)
```

https://github.com/google-research/bert/blob/master/optimization.py

Is NLP Easy? No!

- How to annoy graduate students with four lines of Python code

Is NLP Easy? No!

Witte (2006)

- Ambiguity of a natural language

Ambiguity appears on every analysis level

The classical examples:

- *He saw the man with the telescope.*
- *Time flies like an arrow. Fruit flies like a banana.*

And those are simple...

This does not get better with real-world sentences:

- *The board approved [its acquisition] [by Royal Trustco. Ltd.] [of Toronto] [for $27 a share] [at its monthly meeting].*

(cf. Manning & Schütze)

Is NLP Easy? No!

Witte (2006)

- Complex and subtle relationship between concepts in texts
  - ✓ "AOL merges with Time-Warner"
  - ✓ "Time-Warner is bought by AOL"

- Ambiguity and context sensitivity
  - ✓ automobile = car = vehicle = Hyundai

vs.

Research Trends in NLP

Witte (2006)

- From rule-based approaches to statistical approaches

The classical way: until late 1980's

Rule-based approaches:

- are too rigid for natural language
- suffer from the *knowledge acquisition bottleneck*
- cannot keep up with changing/evolving language
  ex. "to google"

The statistical way: since early 1990's

"Statistical NLP" refers to all quantitative approaches, including Bayes' models, Hidden Markov Models (HMMs), Support Vector Machines (SVMs), Clustering, . . .

- more robust & more flexible
- need a *Corpus* for (supervised or unsupervised) learning

But real-world systems typically combine both.

Research Trends in NLP

Collobert et al. (2011)

- From statistical approaches to machine-learning (deep-learning) approaches

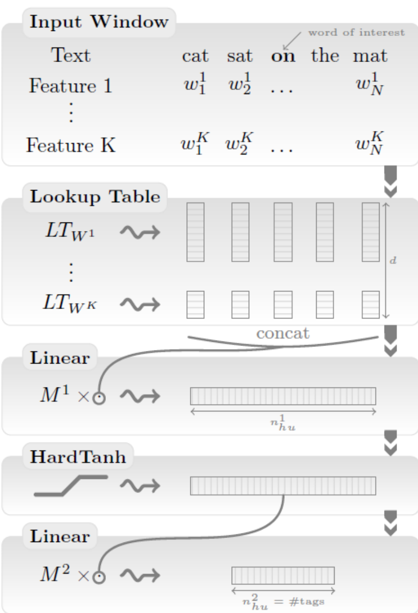


Figure 1: Window approach network.

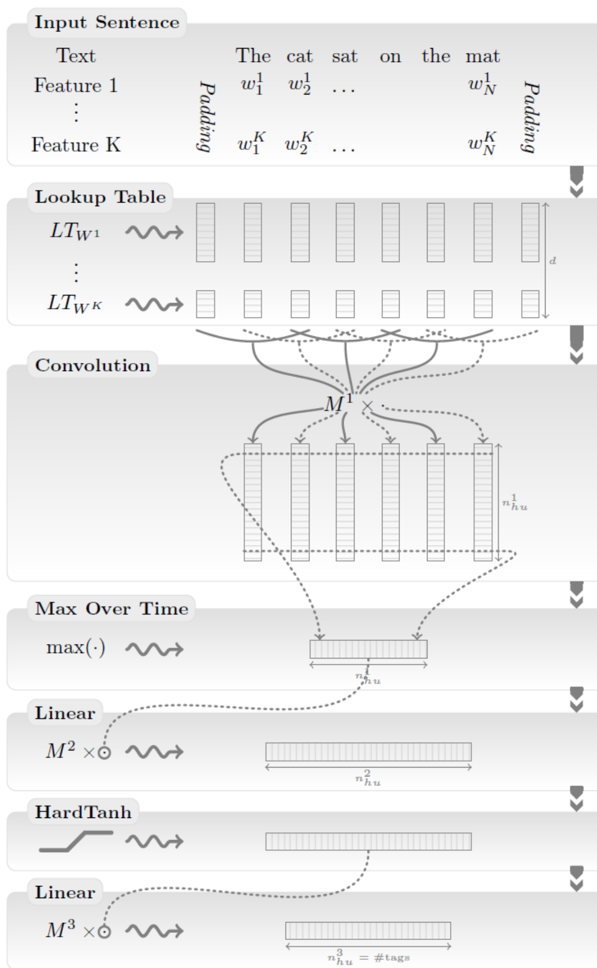


Figure 2: Sentence approach network.

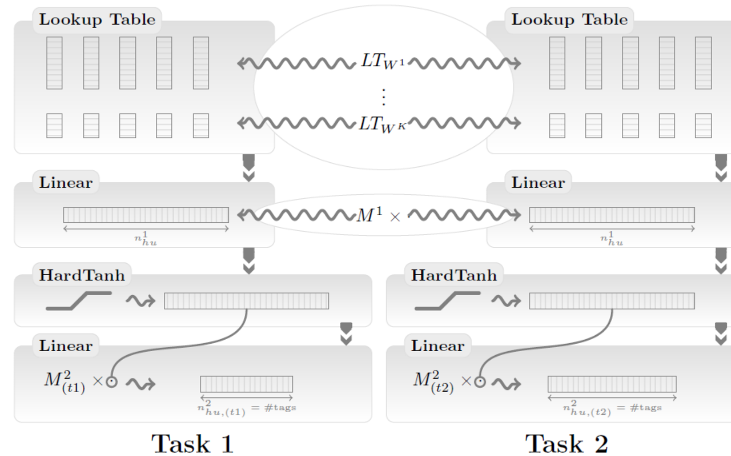
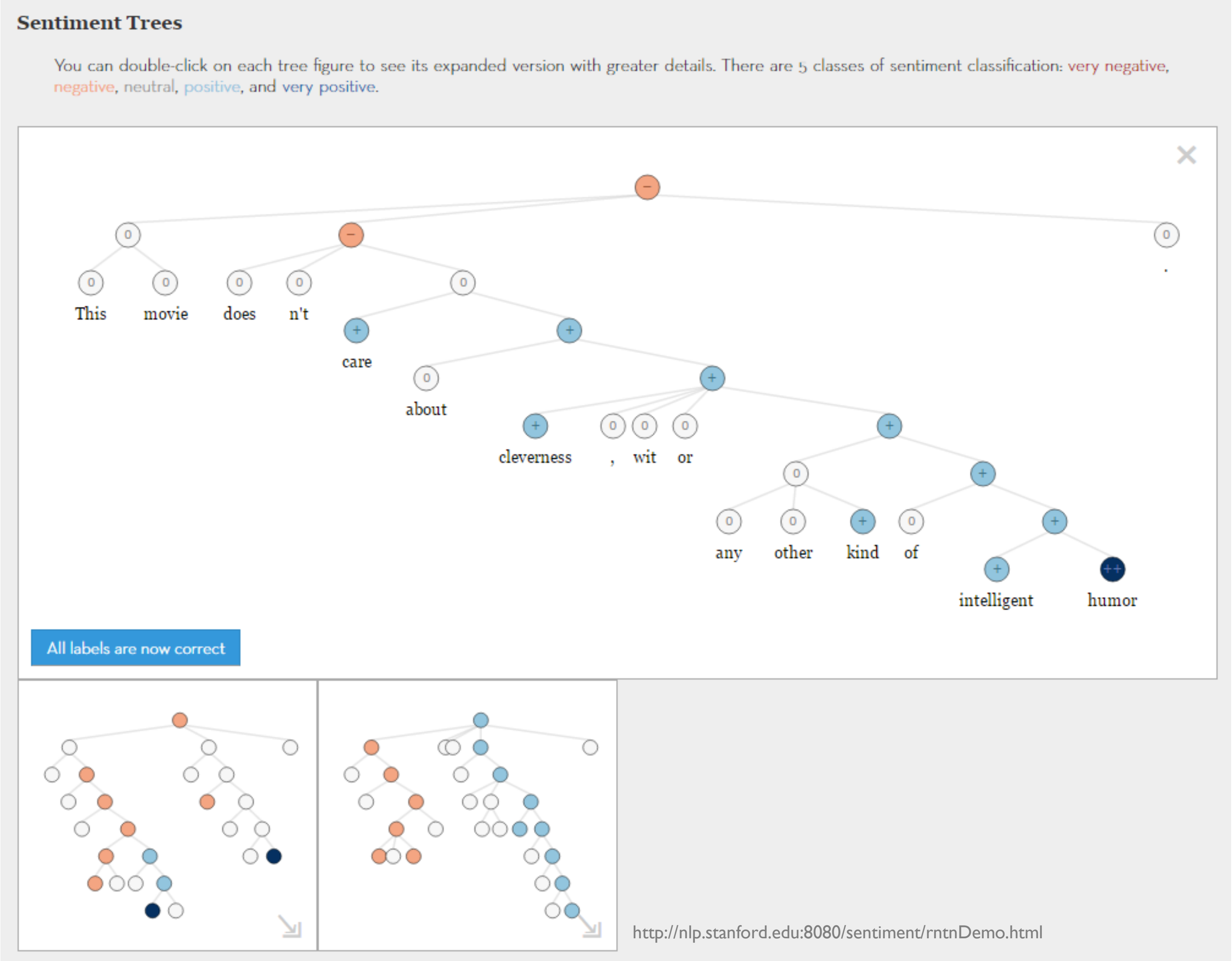


Figure 5: Example of multitasking with NN. Task 1 and Task 2 are two tasks trained with the window approach architecture presented in Figure 1. Lookup tables as well as the first hidden layer are shared. The last layer is task specific. The principle is the same with more than two tasks.

Research Trends in NLP

Socher et al. (2013)

- From statistical approaches to machine-learning (deep-learning) approaches



http://nlp.stanford.edu:8080/sentiment/rntnDemo.html

Research Trends in NLP

Wu et al. (2016)

- From statistical approaches to machine-learning (deep-learning) approaches

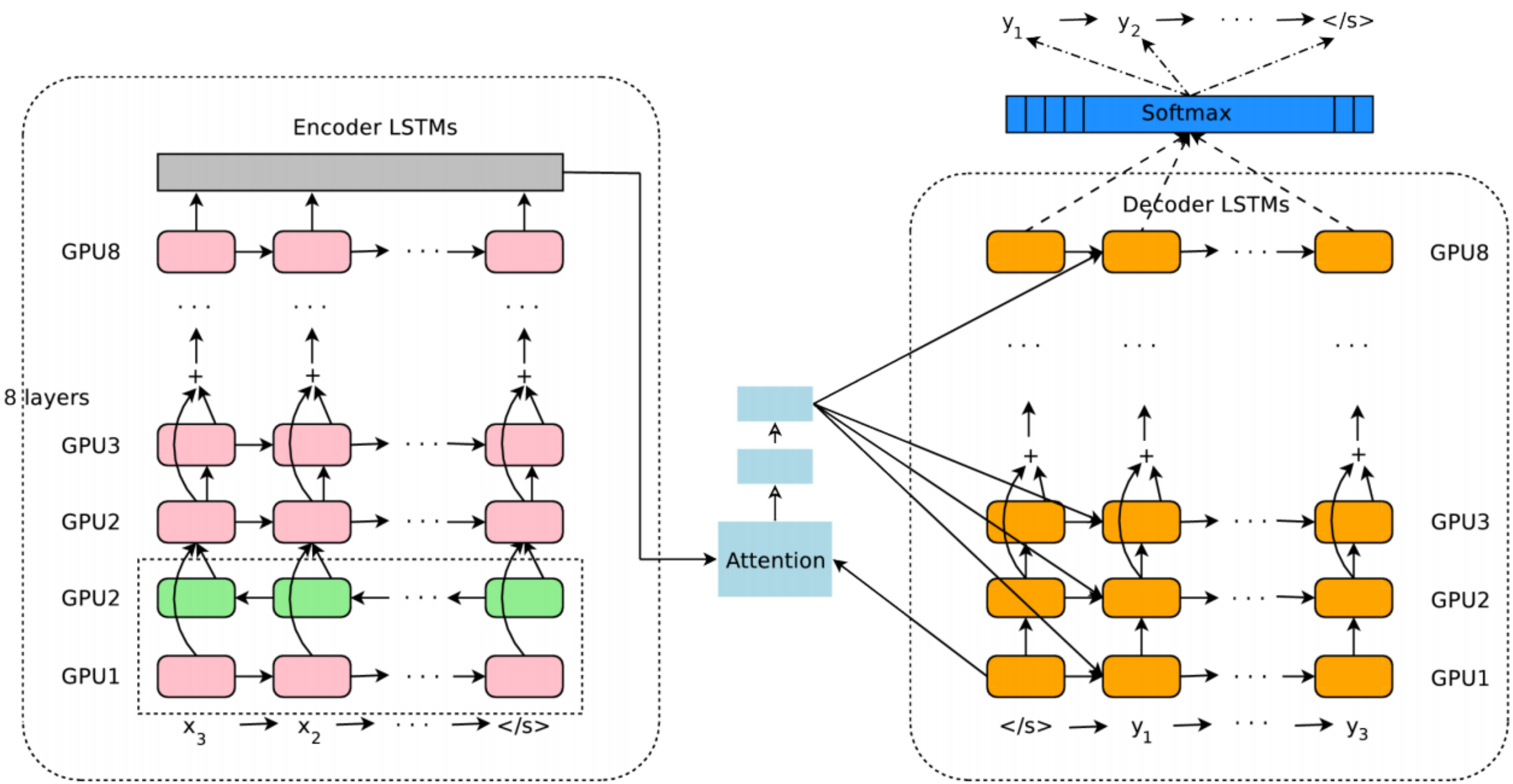
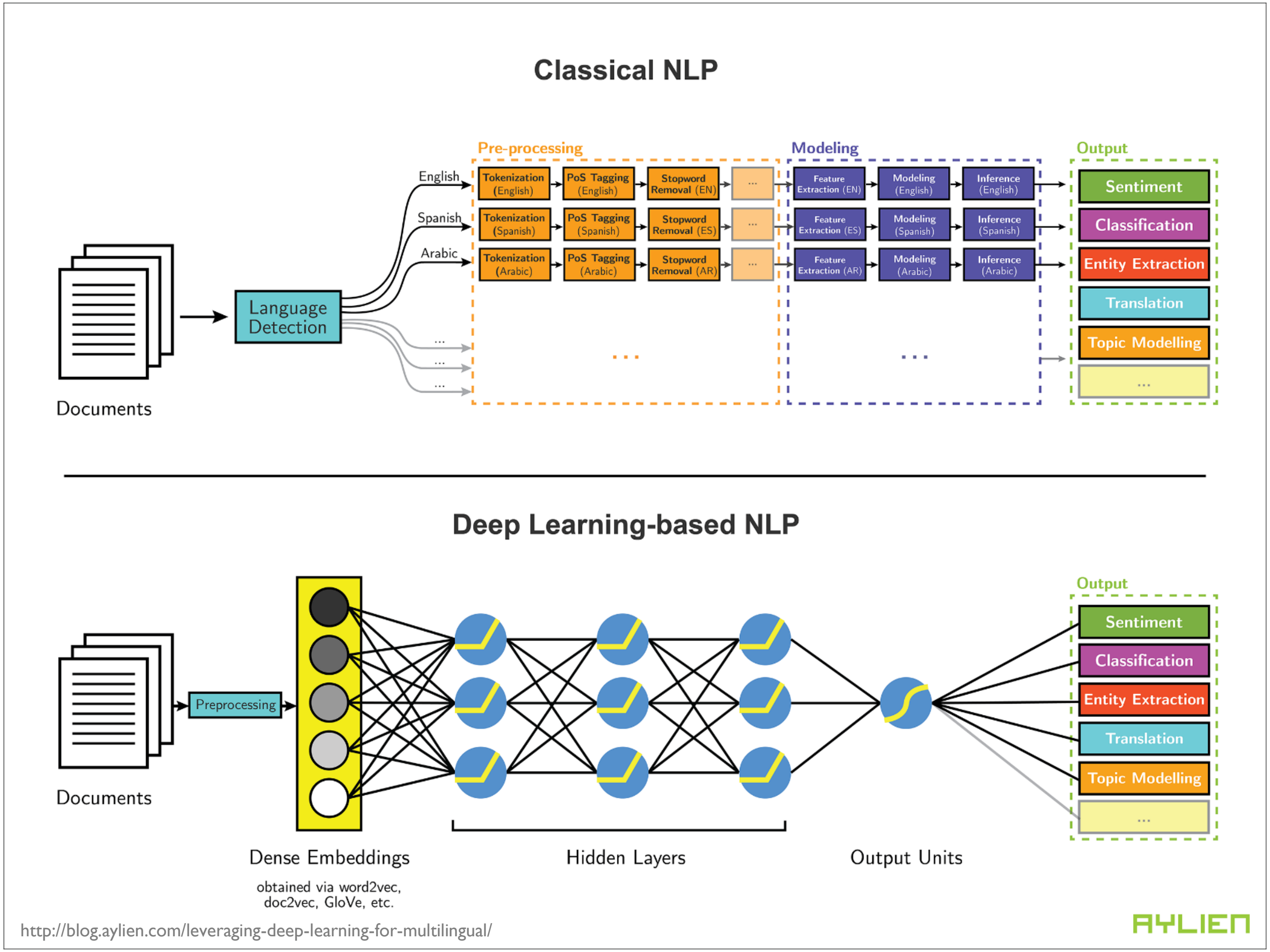


Figure 1: The model architecture of GNMT, Google's Neural Machine Translation system. On the left

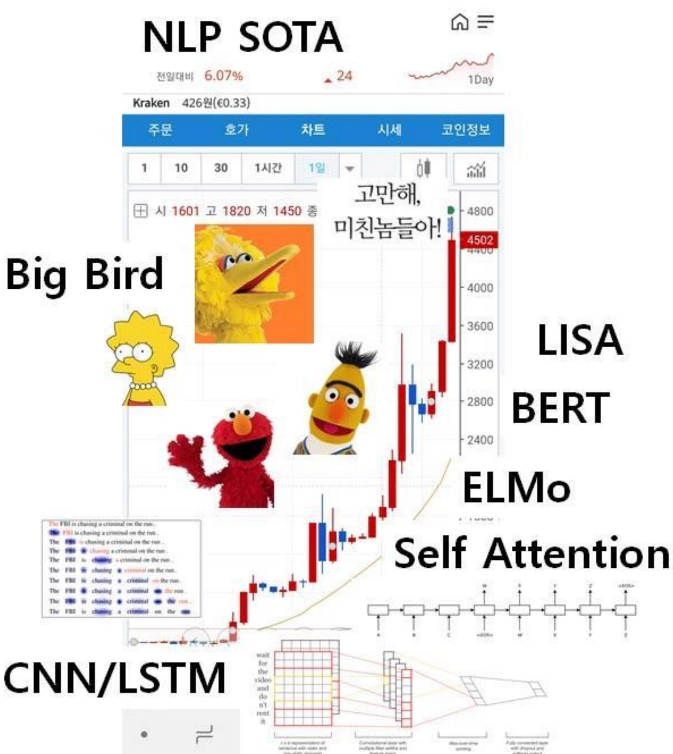
Research Trends in NLP

- End-to-End Multi-Task Learning



Research Trends in NLP

- Performance Improvements



Browse > Natural Language Processing

Natural Language Processing

434 leaderboards • 232 tasks • 100 datasets • 3563 papers with code

Representation Learning

- Representation Learning — 3 leaderboards — 439 papers with code
- Word Embeddings — 427 papers with code
- Graph Embedding — 90 papers with code
- Network Embedding — 54 papers with code
- Sentence Embeddings — 1 leaderboard — 51 papers with code

▸ See all 17 tasks

Machine Translation

- Machine Translation — 44 leaderboards — 562 papers with code
- Transliteration — 16 papers with code
- Unsupervised Machine Translation — 9 leaderboards — 11 papers with code
- Multimodal Machine Translation — 7 papers with code
- Low-Resource Neural Machine Translation — 6 papers with code

▸ See all 6 tasks

Question Answering

- Question Answering — 46 leaderboards — 476 papers with code
- Open-Domain Question Answering — 3 leaderboards — 22 papers with code
- Answer Selection — 2 leaderboards — 19 papers with code
- Community Question Answering — 14 papers with code
- Knowledge Base Question Answering — 1 leaderboard — 8 papers with code

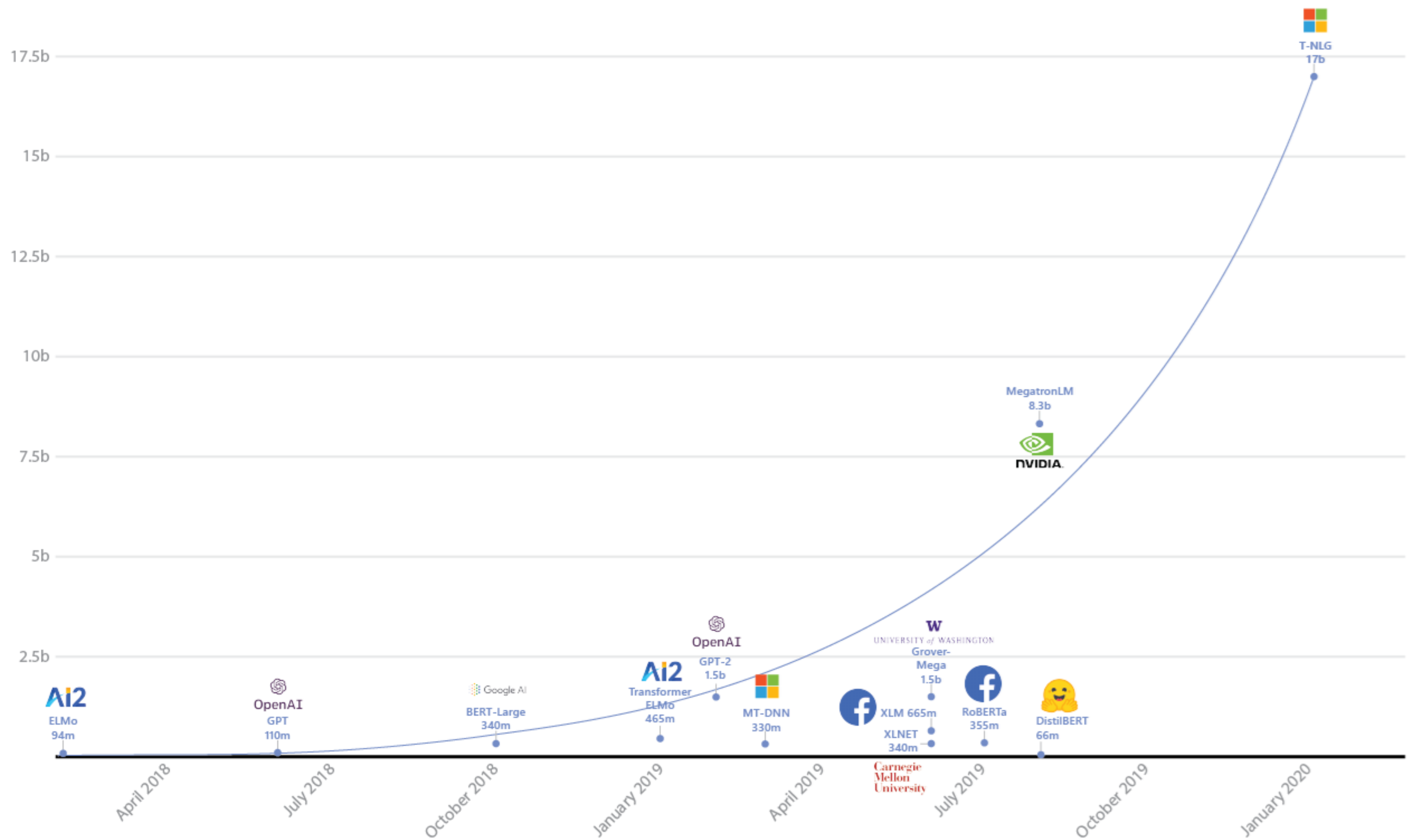
Language Modelling

- Language Modelling — 8 leaderboards — 504 papers with code
- Sentence Pair Modeling — 4 papers with code

https://paperswithcode.com/area/natural-language-processing

Research Trends in NLP

- Performance Improvements with a huge model



https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/

Research Trends in NLP

- An Era of optimization???

수학적 최적화	구분	인공지능
제조·물류·에너지·교통 등 최적의 해결책이 필요한 분야	적용 분야	얼굴·이미지·음성 분석 등 빅데이터 분석 필요한 분야
수학 방정식으로 최적의 답 계산 (연역 추론)	방식	과거 데이터 분석해 유사성 또는 향후 추세 분석(귀납 추론)
-시간·비용이 상대적으로 적게 듦 -새 변수에 바로 대응 가능	장점	-데이터 양 많을수록 정교해짐 -과거에 못 풀던 문제 해결 가능
수학 공식화 어려운 분야에 적용 힘들어	단점	분석 결과가 100% 맞지 않음

MO는 고성능 컴퓨터에 기반해 복잡한 연산을 한다는 점에서 AI와 비슷하다. 하지만 과거의 빅데이터를 분석해 최선의 답을 도출하는 AI와 달리 MO는 수학 공식으로 현재 주어진 한정된 조건에서 가장 이상적인 해결책을 찾아준다. AI가 경험을 바탕으로 가장 나은 방안을 제시한다면, MO는 수학 이론을 동원해 최적 답을 알려주는 것이다. MO는 AI처럼 빅데이터를 분석하는 과정이 필요 없기 때문에 시간·비용이 상대적으로 적게 든다.

Research Trends in NLP

- 10 Exiting ideas of 2018 in NLP (http://ruder.io/10-exciting-ideas-of-2018-in-nlp/)
  - ✓ Unsupervised Machine Translation
  - ✓ Pretrained language models
  - ✓ Common sense inference datasets
  - ✓ Meta-learning
  - ✓ Robust unsupervised methods
  - ✓ Understanding representations
  - ✓ Clever auxiliary tasks
  - ✓ Combining semi-supervised learning with transfer learning
  - ✓ QA and reasoning with large documents
  - ✓ Inductive bias

Research Trends in NLP

- Major NLP Achievements & Papers from 2019
  - ✓ Language Models Are Unsupervised Multitask Learners
  - ✓ XLNet: Generalized Autoregressive Pretraining for Language Understanding
  - ✓ RoBERTa: A Robustly Optimized BERT Pretraining Approach
  - ✓ Emotion-Cause Pair Extraction: A New Task to Emotion Analysis in Texts
  - ✓ Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems
  - ✓ Ordered Neurons: Integrating Tree Structures into Recurrent Neural Networks
  - ✓ Probing the Need for Visual Context in Multimodal Machine Translation
  - ✓ Bridging the Gap between Training and Inference for Neural Machine Translation
  - ✓ On Extractive and Abstractive Neural Document Summarization with Transformer Language Models
  - ✓ CTRL: A Conditional Transformer Language Model For Controllable Generation
  - ✓ ALBERT: A Lite BERT for Self-supervised Learning of Language Representations

https://www.topbots.com/top-ai-nlp-research-papers-2019/

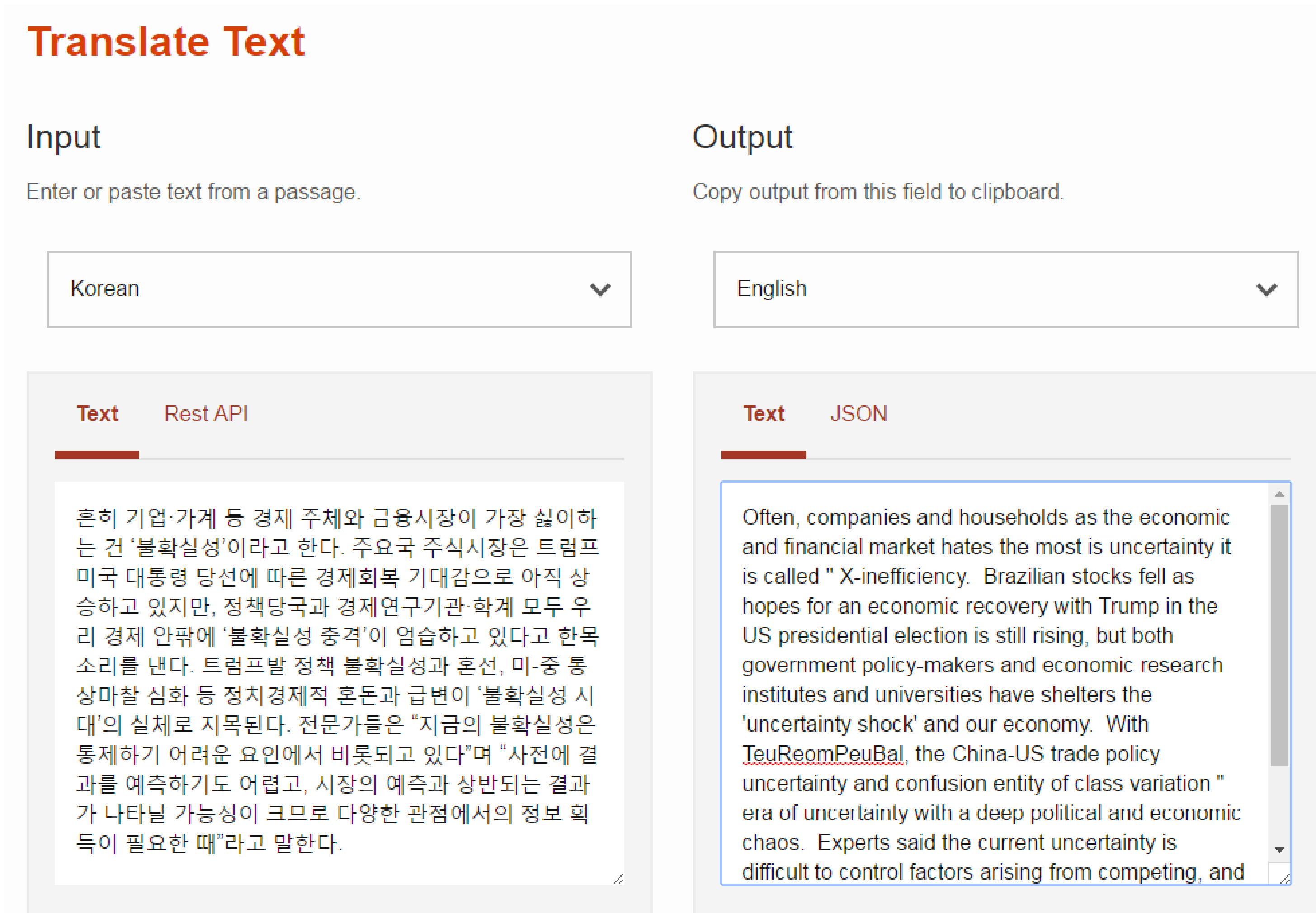
Research Trends in NLP

- 14 NLP research breakthrough you can apply to your business
  - ✓ BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
  - ✓ Sequence Classification with Human Attention
  - ✓ Phrase-Based & Neural Unsupervised Machine Translation
  - ✓ What you can cram into a single vector: Probing sentence embeddings for linguistic properties
  - ✓ SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference
  - ✓ Deep contextualized word representations
  - ✓ Meta-Learning for Low-Resource Neural Machine Translation
  - ✓ Linguistically-Informed Self-Attention for Semantic Role Labeling
  - ✓ A Hierarchical Multi-task Approach for Learning Embeddings from Semantic Tasks
  - ✓ Know What You Don't Know: Unanswerable Questions for SQuAD
  - ✓ An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling
  - ✓ Universal Language Model Fine-tuning for Text Classification
  - ✓ Improving Language Understanding by Generative Pre-Training
  - ✓ Dissecting Contextual Word Embeddings: Architecture and Representation

(https://www.topbots.com/most-important-ai-nlp-research/?utm_campaign=meetedgar&utm_medium=social&utm_source=meetedgar.com)

Research Trends in NLP

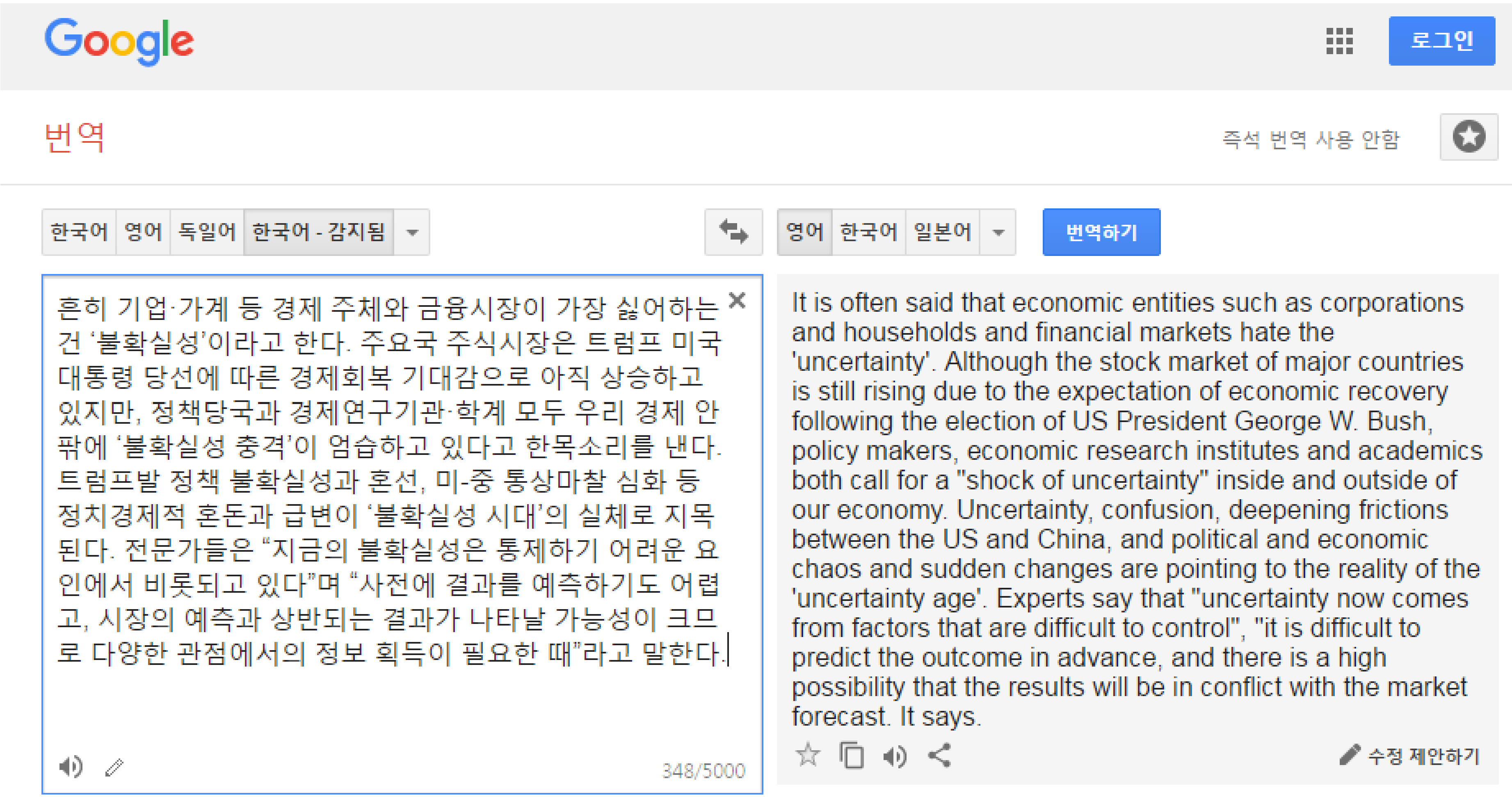
- Statistical translation vs. deep learning-based translation



https://language-translator-demo.mybluemix.net/

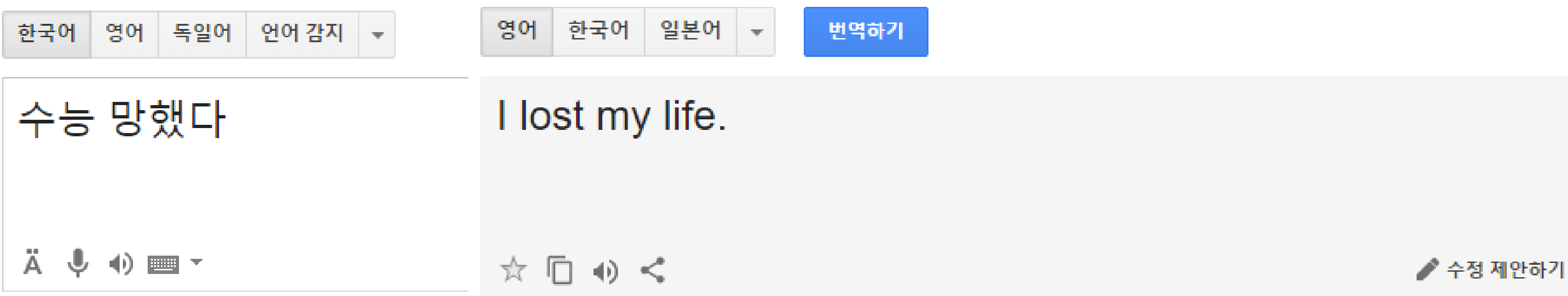
Research Trends in NLP

- Statistical translation vs. deep learning-based translation

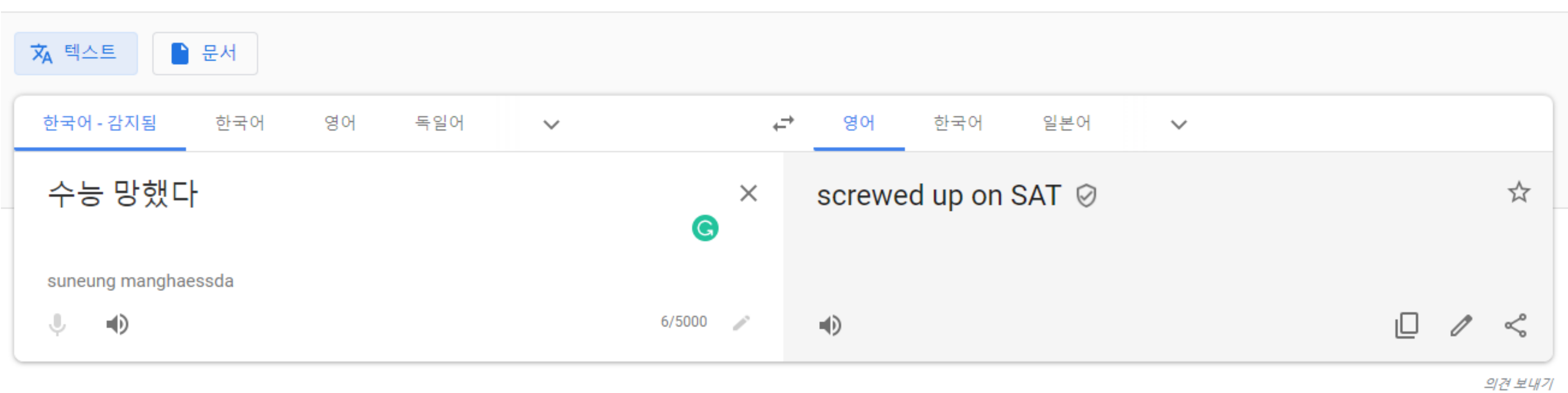


Research Trends in NLP

- Provide your inputs to improve the machine translator!

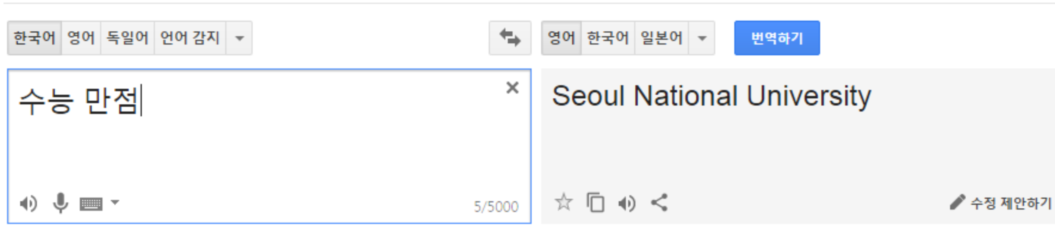


- 2018. 03. 06 & 2019. 03. 02 & 2020. 03. 02

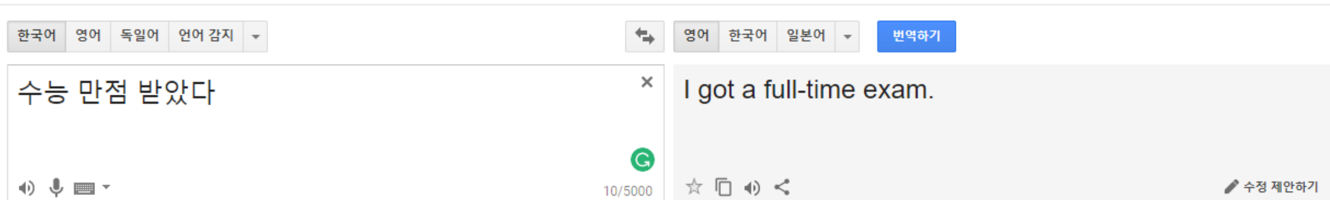


Research Trends in NLP

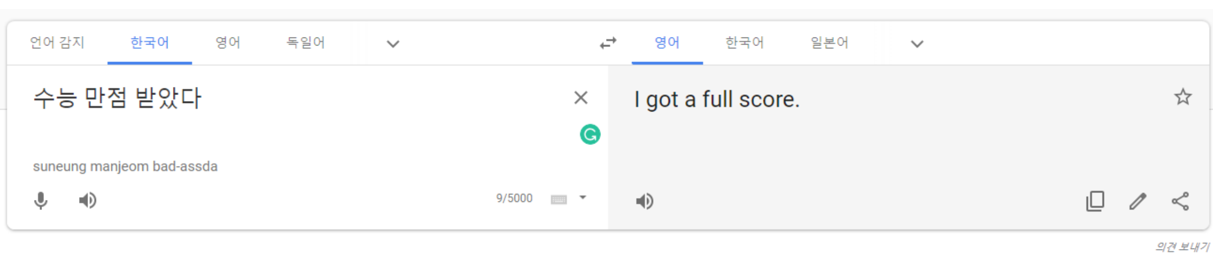
- Provide your inputs to improve the machine translator!



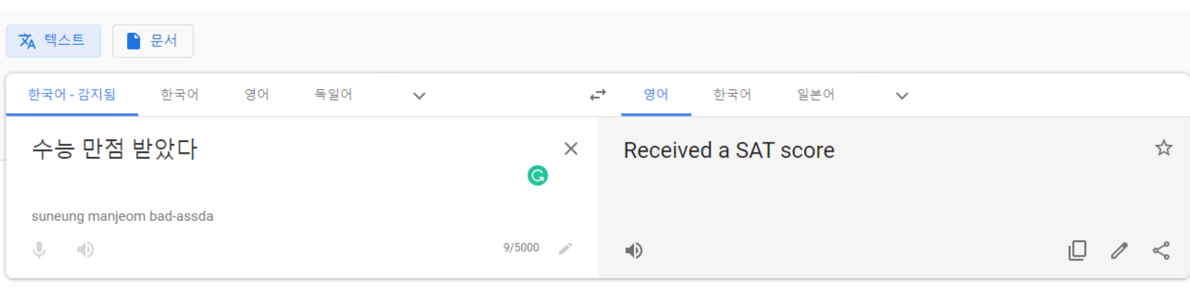
- 2018. 03. 06



- 2019. 03. 02



- 2020. 03. 02



Data Quality in NLP

- ExoBrain Project

학습 모델 및 데이터 목록

한국어 BERT 언어모델 | 언어처리 학습데이터 | 음성 학습데이터 | 객체검출 학습데이터

언어처리 학습데이터

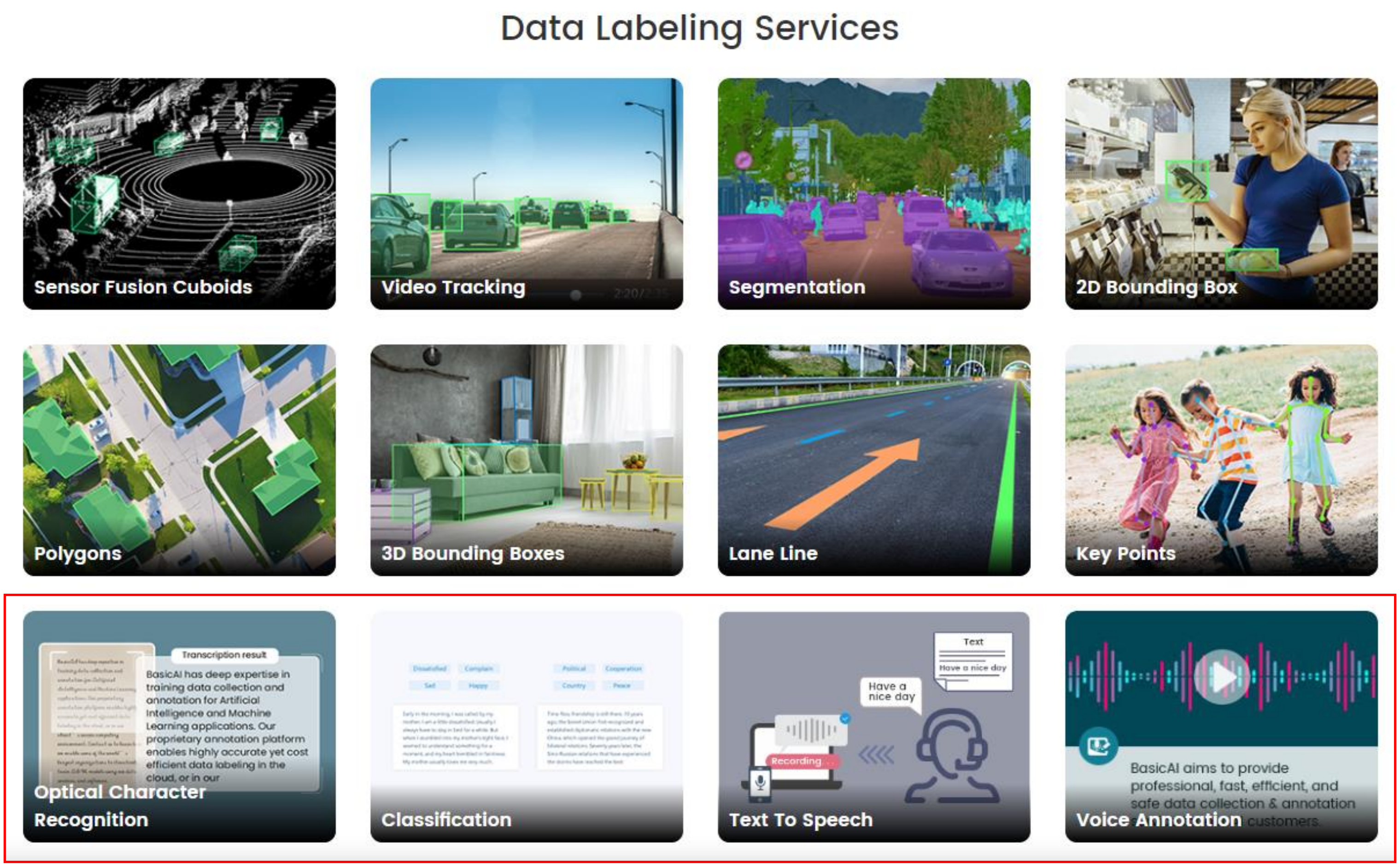
한국어 분석 및 질의응답 기술을 개발하기 위한 과학기술정보통신부 소프트웨어 분야 R&D인 엑소브레인 과제에서는 다양한 지식산업 환경에서 전문가 수준의 질의응답 서비스의 제공을 위하여 ETRI, 울산대, KAIST, 충북대, 강원대 등 국내 여러 연구기관이 힘을 합쳐 연구하고 있습니다.

엑소브레인 과제를 수행하면서 ETRI와 함께 각 연구기관에서 구축한 언어처리 학습데이터(엑소브레인 말뭉치 v4.0)를 공개하여 유사 분야 연구에 도움이 되고자 합니다. 공개하는 엑소브레인 말뭉치 v4.0은 아래와 같이 구성되어 있습니다.

엑소브레인 QA Datasets (ETRI)	퀴즈 QA Datasets ▪ 퀴즈 분야 질의응답을 위한 4개 유형 (객관식/주관식/가부형/연상형)의 퀴즈 QA datasets (569개)
	SQuAD 한국어 QA Dataset ▪ SQuAD 질문의 위키피디아 한국어 번역 QA datasets (표준태깅, 339개) MRC 한국어 QA Dataset ▪ 한국어 위키피디아를 대상으로 구축한 MRC(Machine Reading Comprehension) QA datasets(10,000개)
	위키피디아 단문질문 QA Datasets ▪ 상/중/하 난이도 별 패러프레이즈 QA datasets(표준태깅, 300개) ▪ 일반상식 분야 QA dataset(기본 태깅, 1,776개)
엑소브레인 언어분석 말뭉치 (ETRI/강원대)	언어분석 통합 말뭉치 ▪ 언어분석 6개 기술(형태소분석, 다의어 어휘의미분석, 세분류 개체명인식, 의존구문분석, 의미역인식, 상호참조해결)의 태깅 가이드라인과 자연어 질의응답을 위한 질문/정답 포맷의 뉴스기사 대상 태깅 말뭉치 (2,593문장, 33,131어절)
	세부기술 별 말뭉치 ▪ 개체명 인식 태깅 가이드라인 및 말뭉치 (인명/장소/조직/날짜/시간 5개 태그, 10,000 문장) ▪ 의미역 인식 태깅 가이드라인 및 말뭉치 (625문장, 7,436어절) ▪ 의존구문분석 태깅 가이드라인 및 말뭉치 (2,225문장, 27,317어절) TTA 공인인증 말뭉치 (추후제공예정) ▪ 엑소브레인 과제 결과물의 객관적인 성능 측정을 위해 2018년 한국정보통신기술협회(TTA)에서 시행한 공인인증 평가에 사용된 평가셋(개체명 인식: 459문장, 의미역 인식(필수역 대상): 450문장)
UCorpus-HG 말뭉치 (울산대학교)	울산대학교 형태/의미 말뭉치(UCorpus-HG) ▪ 표준국어대사전 기반 모든 동형이의어 대상으로 어깨번호를 부착한 말뭉치 ▪ 원문: 세종 형태의미 말뭉치, 신문, 초등학교 국어교과서, 법률, 사전뜻풀이/용례 ▪ 전체 1,909,840 문장, 18,869,517 어절 (학습 말뭉치 90%, 평가 말뭉치 10%로 분리 제공)
엑소브레인 Korean TimeBank 및 SpaceBank (KAIST/충북대)	한국어 시간 정보 주석 말뭉치: Korean TimeBank (KAIST) ▪ 한국어 시간 정보 자동 추출을 위한 한국어 시간 정보 주석 가이드라인 및 말뭉치 ▪ 말뭉치 구성: 812 문서, 5,467 문장 태깅 (시간개체 4,509개, 시간관계 5,182개)
	한국어 공간 정보 주석 말뭉치: Korean SpaceBank (충북대학교) ▪ 한국어 공간 정보 자동 추출을 위한 한국어 공간 정보 주석 가이드라인 및 말뭉치 ▪ 말뭉치 구성 : 개체 태그 7종류, 관계 태그 4종류, 2,264 문장 태깅
엑소브레인 패러프레이즈 말뭉치 (KAIST)	한국어 패러프레이즈 말뭉치: Korean Paraphrase Corpus(KAIST) ▪ 한국어 패러프레이즈 인식 및 평가를 위한 주석 가이드라인 및 말뭉치 ▪ 말뭉치 구성: 패러프레이즈 관계 2,000문장 쌍과 출처, 유사도(0-5)/난이도(상/중/하) 표준 태깅, 의미(실질) 형태소 정보 태깅

Data Quality in NLP

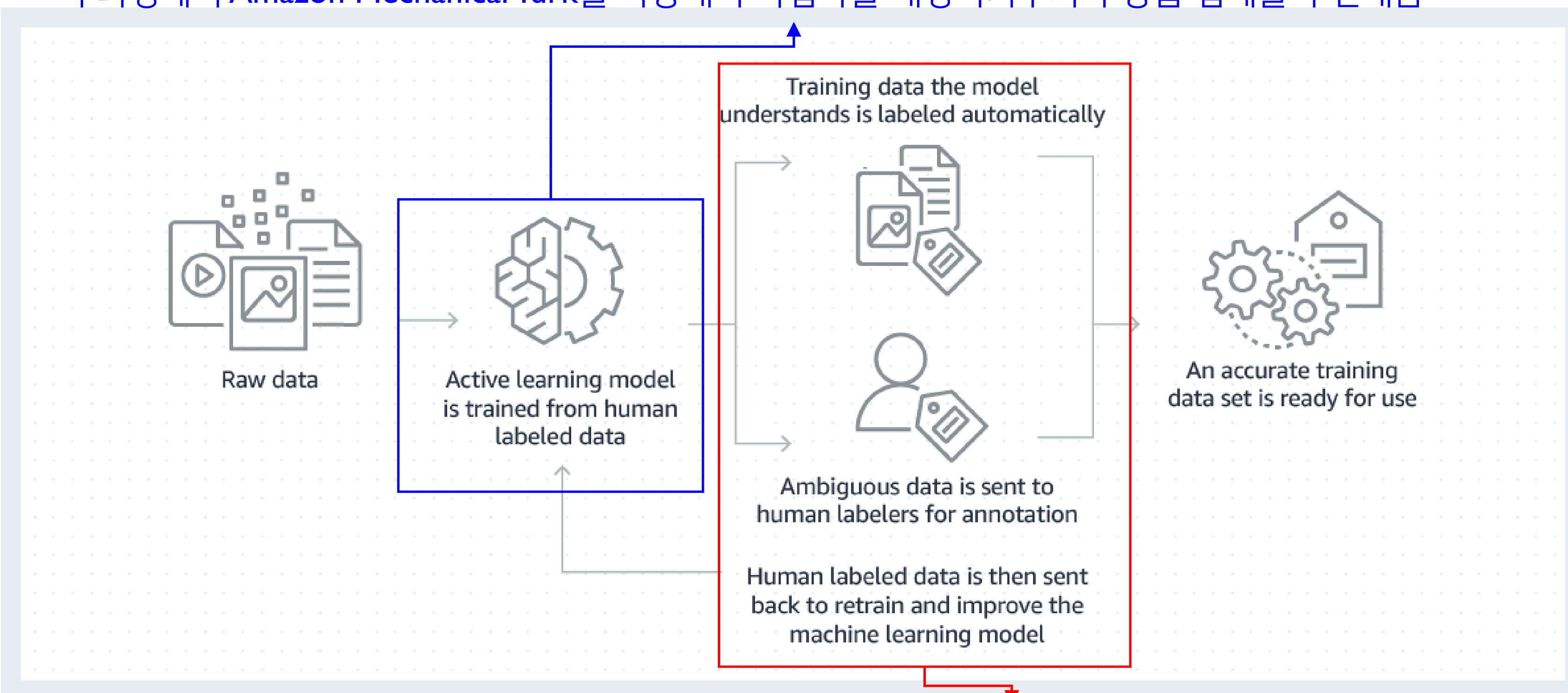
- Data Annotation as a Business Model
  - ✓ Scale AI: https://scale.com/
  - ✓ Basic AI: https://www.basic.ai/



Data Quality in NLP

- Data Annotation as a Business Model
  - ✓ Amazon SageMaker Ground Truth: https://aws.amazon.com/ko/sagemaker/groundtruth/
    - Data labeling Platform

처음에는 사람에 의해 labeling 작업 수행
이 과정에서 Amazon Mechanical Turk를 사용해서 작업자를 매칭시켜주거나 공급 업체를 추천해줌



1차 레이블링된 데이터를 이용해서 AI 모델을 학습시킨 후,
모델의 신뢰도가 낮을 경우에 사람에게 확인 요청을 하는 feedback loop를 거침

Data Quality in NLP

- Data Annotation as a Business Model (Social Enterprise)
  - ✓ DataMaker: https://www.rdproject.kr/#section-service
  - ✓ 테스트웍스: http://www.testworks.co.kr/

IMAGE AUDIO

Audio Slicing | Audio to Text | Advanced Audio Labeling

Audio Slicing은 음원 파일을 조건에 따라 분리하는 작업입니다.
음성 AI 서비스 개발 시, 화자에 따른 음원 분리, 노이즈 구간 제거 등을 할 수 있어
이를 토대로 AI를 학습시키는 데 활용될 수 있습니다.

플랫폼 특장점 3

**이중 전수 검수를 통한
고품질의 데이터 보장**

저희는 아프리카 가나 데이터 랩에서 1차 작업 및 전수 검수,
대한민국 데이터 랩에서 전담 검수자들이 2차 전수 검수를 시행합니다.

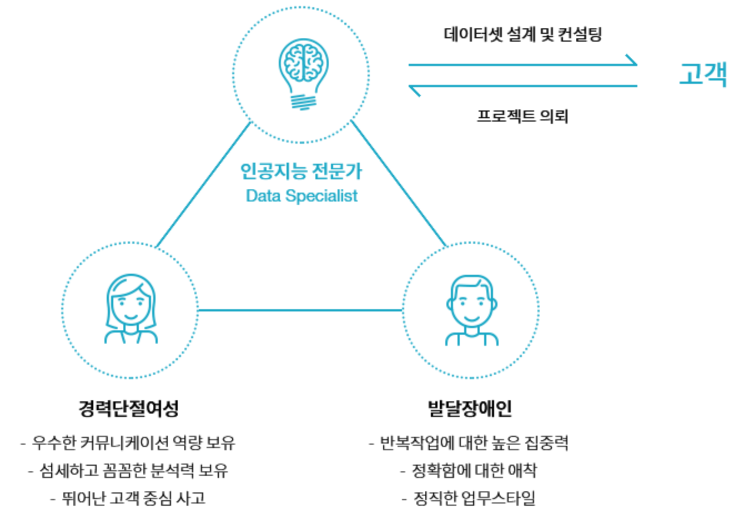
모든 작업자들은 보상금을 지급받기 위해서 데이터메이커의 엄격한 통과 기준을
준수해야 합니다. 저희는 2가지 방식의 검수 시스템을 보유하고 있으며,
의뢰자님이 검수 옵션을 선택하실 수 있습니다.

데이터 가공

인공지능 알고리즘 개발을 위해 학습/검증용 데이터셋 구축 서비스를 제공

인공지능 데이터셋 구축 전문 인력

인공지능 전문가의 리드 하에, 데이터 가공 작업에 풍부한 경험과
특장점을 가진 인원들이 프로젝트에 투입됩니다.



AGENDA

Lexical Analysis

- Goals of lexical analysis
  - ✓ Convert a sequence of characters into a sequence of tokens, i.e., meaningful character strings.
    - In natural language processing, morpheme is a basic unit
    - In text mining, word is commonly used as a basic unit for analysis
- Process of lexical analysis
  - ✓ Tokenizing
  - ✓ Part-of-Speech (POS) tagging
  - ✓ Additional analysis: named entity recognition (NER), noun phrase recognition, sentence split, chunking, etc.

Lexical Analysis

Hirschberg and Manning (2015)

- Examples of Linguistic Structure Analysis

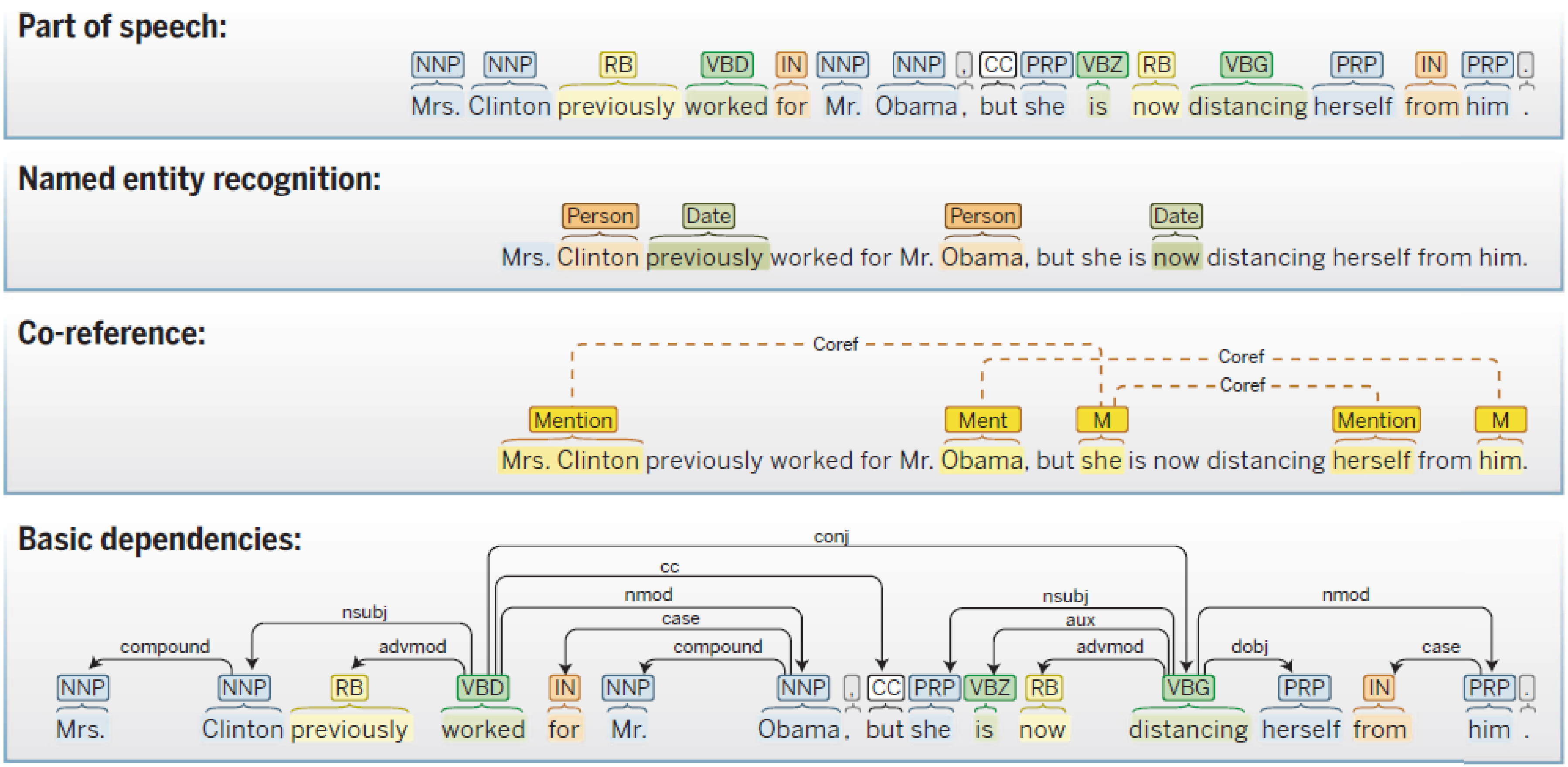


Fig. 1. Many language technology tools start by doing linguistic structure analysis. Here we show output from Stanford CoreNLP. As shown from top to bottom, this tool determines the parts of speech of each word, tags various words or phrases as semantic named entities of various sorts, determines which entity mentions co-refer to the same person or organization, and then works out the syntactic structure of each sentence, using a dependency grammar analysis.

Lexical Analysis I: Sentence Splitting

Witte (2016)

- Sentence is very important in NLP, but it is not critical for some Text Mining tasks

Mark Sentence Boundaries

Detects sentence units. Easy case:

- often, sentences end with ".", "!", or "?"

Hard (or annoying) cases:

- difficult when a "." do not indicate an EOS:
  "MR. X", "3.14", "Y Corp.", . . .
- we can detect common abbreviations ("U.S."), but what if a sentence ends with one?
  ". . .announced today by the U.S. The. . .
- Sentences can be *nested* (e.g., within quotes)

Correct sentence boundary is important

for many downstream analysis tasks:

- POS-Taggers maximize probabilites of tags within a sentence
- Summarization systems rely on correct detection of sentence

Lexical Analysis 2: Tokenization

- Text is split into basic units called Tokens
  - ✓ word tokens, number tokens, space tokens, …

```
> crude[[1]]
<<PlainTextDocument (metadata: 15)>>
Diamond Shamrock Corp said that
effective today it had cut its contract prices for crude oil by
1.50 dlrs a barrel.
    The reduction brings its posted price for West Texas
Intermediate to 16.00 dlrs a barrel, the copany said.
    "The price reduction today was made in the light of falling
oil product prices and a weak crude oil market," a company
spokeswoman said.
    Diamond is the latest in a line of U.S. oil companies that
have cut its contract, or posted, prices over the last two days
citing weak oil markets.
 Reuter
```

```
> MC_tokenizer(crude[[1]])
  [1] "Diamond"     "Shamrock"    "Corp"        "said"        "that"
  [6] "effective"   "today"       "it"          "had"         "cut"
 [11] "its"         "contract"    "prices"      "for"         "crude"
 [16] "oil"         "by"          ""            ""            ""
 [21] ""            ""            "dlrs"        "a"           "barrel"
 [26] ""            ""            ""            ""            ""
 [31] "The"         "reduction"   "brings"      "its"         "posted"
 [36] "price"       "for"         "West"        "Texas"       "Intermediate"
 [41] "to"          ""            ""            ""            ""
 [46] ""            ""            "dlrs"        "a"           "barrel"
 [51] ""            "the"         "copany"      "said"        ""
 [56] ""            ""            ""            ""            ""
 [61] "The"         "price"       "reduction"   "today"       "was"
 [66] "made"        "in"          "the"         "light"       "of"
 [71] "falling"     "oil"         "product"     "prices"      "and"
 [76] "a"           "weak"        "crude"       "oil"         "market"
 [81] ""            ""            "a"           "company"     "spokeswoman"
 [86] "said"        ""            ""            ""            ""
 [91] ""            "Diamond"     "is"          "the"         "latest"
 [96] "in"          "a"           "line"        "of"          "U"
[101] "S"           ""            "oil"         "companies"   "that"
[106] "have"        "cut"         "its"         "contract"    ""
[111] "or"          "posted"      ""            "prices"      "over"
[116] "the"         "last"        "two"         "days"        "citing"
[121] "weak"        "oil"         "markets"     ""            ""
[126] "Reuter"
```

```
> scan_tokenizer(crude[[1]])
 [1] "Diamond"      "Shamrock"    "Corp"        "said"       "that"
 [6] "effective"    "today"       "it"          "had"        "cut"
[11] "its"          "contract"    "prices"      "for"        "crude"
[16] "oil"          "by"          "1.50"        "dlrs"       "a"
[21] "barrel."      "The"         "reduction"   "brings"     "its"
[26] "posted"       "price"       "for"         "West"       "Texas"
[31] "Intermediate" "to"          "16.00"       "dlrs"       "a"
[36] "barrel,"      "the"         "copany"      "said."      "\"The"
[41] "price"        "reduction"   "today"       "was"        "made"
[46] "in"           "the"         "light"       "of"         "falling"
[51] "oil"          "product"     "prices"      "and"        "a"
[56] "weak"         "crude"       "oil"         "market,\""  "a"
[61] "company"      "spokeswoman" "said."       "Diamond"    "is"
[66] "the"          "latest"      "in"          "a"          "line"
[71] "of"           "U.S."        "oil"         "companies"  "that"
[76] "have"         "cut"         "its"         "contract,"  "or"
[81] "posted,"      "prices"      "over"        "the"        "last"
[86] "two"          "days"        "citing"      "weak"       "oil"
[91] "markets."     "Reuter"
```

	MC	Scan
Space	Not removed	Removed
Punctuation	Removed	Not removed
Numbers	Removed	Not removed
Special characters	Removed	Not removed

Lexical Analysis 2: Tokenization

- Even tokenization can be difficult
  - ✓ Is John's sick one token or two?
    - If one → problems in parsing (where is the verb?)
    - If two → what do we do with John's house?
  - ✓ What to do with hyphens?
    - database vs. data-base vs. data base
  - ✓ What to do with "C++", "A/C", ":-)", "…", "ㅋㅋㅋㅋㅋㅋㅋㅋ"?
  - ✓ Some languages do not use whitespace (e.g., Chinese)

> 2013年5月，习主席在视察成都战区时，郑重提出在适当时候召开全军政治工作会议，并明确提出到古田召开这次会议，以更好弘扬我党我军的光荣传统和优良作风。6月，总政治部向中央军委提交《关于筹备召开全军政治工作会议的请示》，提出要通过召开会议形成一个指导性文件。习主席随即批示同意，明确要求这个文件要充分体现深厚的历史积淀和政治意蕴，能够管一个时期，起到历史性作用。

- Consistent tokenization is important for all later processing steps.

Lexical Analysis 3: Morphological Analysis

- Morphological Variants: Stemming and Lemmatization

Morphological Variants

Words are changed through a morphological process called *inflection*:

- typically indicates changes in case, gender, number, tense, etc.
- example *car* $\longrightarrow$ cars, *give* $\longrightarrow$ *gives, gave, given*

Goal: "normalize" words

Stemming and Lemmatization

Two main approaches to normalization:

Stemming reduce words to a *base form*

Lemmatization reduce words to their *lemma*

Main difference: stemming just finds **any** base form, which doesn't even need to be a word in the language! Lemmatization find the actual *root* of a word, but requires morphological analysis.

Lexical Analysis 3: Morphological Analysis

- Stemming

Stemming

Commonly used in Information Retrieval:

- Can be achieved with rule-based algorithms, usually based on suffix-stripping
- Standard algorithm for English: the *Porter* stemmer
- Advantages: simple & fast
- Disadvantages:
  - Rules are language-dependent
  - Can create words that do not exist in the language, e.g.,
    computers $\longrightarrow$ *comput*
  - Often reduces different words to the same stem, e.g.,
    army, arm $\longrightarrow$ *arm*
    stocks, stockings $\longrightarrow$ *stock*
- Stemming for German: German stemmer in the full-text search engine *Lucene*, *Snowball* stemmer with German rule file

Lexical Analysis 3: Morphological Analysis

Witte (2016)

- Lemmatization

Lemmatization

Lemmatization is the process of deriving the base form, or *lemma*, of a word from one of its inflected forms. This requires a morphological analysis, which in turn typically requires a *lexicon*.

- Advantages:
  - identifies the *lemma* (root form), which is an actual word
  - less errors than in stemming
- Disadvantages:
  - more complex than stemming, slower
  - requires additional language-dependent resources
- While stemming is good enough for Information Retrieval, Text Mining often requires lemmatization
  - Semantics is more important (we need to distinguish an *army* and an *arm*!)
  - Errors in low-level components can multiply when running downstream

Lexical Analysis 3: Morphological Analysis

- Stemming vs. Lemmatization

Word	Stemming	Lemmatization
Love	Lov	Love
Loves	Lov	Love
Loved	Lov	Love
Loving	Lov	Love
Innovation	Innovat	Innovation
Innovations	Innovat	Innovation
Innovate	Innovat	Innovate
Innovates	Innovat	Innovate
Innovative	Innovat	Innovative

Lexical Analysis 3: Morphological Analysis

- Stemming vs. Lemmatization with crude example

```
> crude[[1]]
<<PlainTextDocument (metadata: 15)>>
Diamond Shamrock Corp said that
effective today it had cut its contract prices for crude oil by
1.50 dlrs a barrel.
    The reduction brings its posted price for West Texas
Intermediate to 16.00 dlrs a barrel, the copany said.
    "The price reduction today was made in the light of falling
oil product prices and a weak crude oil market," a company
spokeswoman said.
    Diamond is the latest in a line of U.S. oil companies that
have cut its contract, or posted, prices over the last two days
citing weak oil markets.
 Reuter
```

Stemming

```
> stemCorpus[[1]]
<<PlainTextDocument (metadata: 7)>>
diamond shamrock corp said that
effect today it had cut it contract price for crude oil by
 dlrs a barrel
    the reduct bring it post price for west texas
intermedi to  dlrs a barrel the copani said
    the price reduct today was made in the light of falling
oil product price and a weak crude oil market a company
spokeswoman said
    diamond is the latest in a line of us oil compani that
hav cut it contract or post price over the last two days
cit weak oil markets
 reuter
```

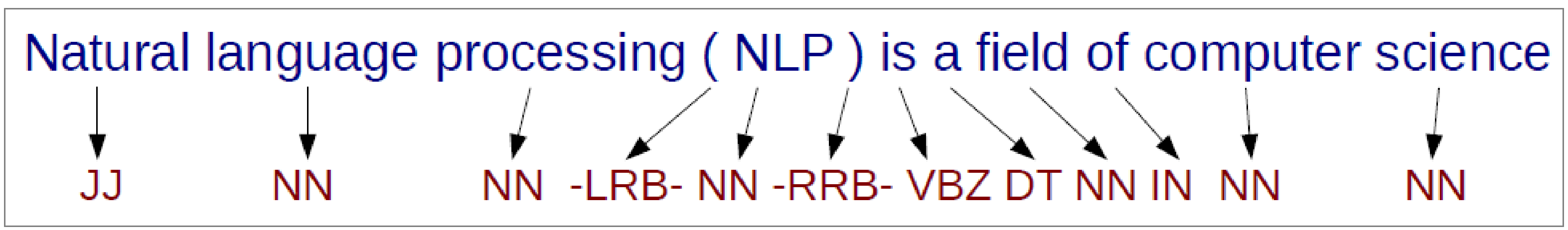
Lemmatization

```
> LemmaCorpus1
[1] "diamond shamrock corp say that effective today it have
cut it contract price for crude oil by dlr a barrel the redu
ction bring it post price for w texa intermediate to dlr a b
arrel the copany say the price reduction today be make in th
e light have fall oil product price and a weak crude oil mar
ket a company spokeswoman say diamond be the late in a line
have us oil company that have cut it contract or post price
ov the last two day cite weak oil market reut"
```

Lexical Analysis 4: Part-of-Speech (POS) Tagging

Witte (2016)

- Part of speech (POS) tagging
  - ✓ Given a sentence X, predict its part of speech sequence Y
    - ▪ Input: tokens that sentence may have ambiguity
    - ▪ Output: most appropriate tag by considering its definition and contexts (relationship with adjacent and related words in phrases, sentence, or paragraph)
  - ✓ A type of "structured" prediction



- Different POS tags for the same token
  - ✓ I <u>love</u> you. → "love" is a verb
  - ✓ All you need is <u>love</u>. → "love" is noun

Lexical Analysis 4: Part-of-Speech (POS) Tagging

- POS Tagging

POS-Tagging

A statistical POS Tagger scans tokens and assigns POS Tags.

A black cat plays... $\rightarrow$ *A/DT black/JJ cat/NN plays/VB...*

- relies on different word order probabilities
- needs a manually tagged corpus for machine learning

Note: *this is not parsing!*

Lexical Analysis 4: Part-of-Speech (POS) Tagging

- Tagsets: English

Penn Treebank

TAG	DESCRIPTION	EXAMPLE
CC	conjunction, coordinating	*and, or, but*
CD	cardinal number	*five, three, 13%*
DT	determiner	*the, a, these*
EX	existential there	*there were six boys*
FW	foreign word	*mais*
IN	conjunction, subordinating or preposition	*of, on, before, unless*
JJ	adjective	*nice, easy*
JJR	adjective, comparative	*nicer, easier*
JJS	adjective, superlative	*nicest, easiest*
LS	list item marker	
MD	verb, modal auxillary	*may, should*
NN	noun, singular or mass	*tiger, chair, laughter*
NNS	noun, plural	*tigers, chairs, insects*
NNP	noun, proper singular	*Germany, God, Alice*
NNPS	noun, proper plural	*we met two Christmases ago*
PDT	predeterminer	*both his children*
POS	possessive ending	*'s*
PRP	pronoun, personal	*me, you, it*
PRP$	pronoun, possessive	*my, your, our*
RB	adverb	*extremely, loudly, hard*
RBR	adverb, comparative	*better*
RBS	adverb, superlative	*best*
RP	adverb, particle	*about, off, up*
SYM	symbol	*%*
TO	infinitival to	*what to do?*
UH	interjection	*oh, oops, gosh*
VB	verb, base form	*think*
VBZ	verb, 3rd person singular present	*she thinks*
VBP	verb, non-3rd person singular present	*I think*
VBD	verb, past tense	*they thought*
VBN	verb, past participle	*a sunken ship*
VBG	verb, gerund or present participle	*thinking is fun*
WDT	*wh*-determiner	*which, whatever, whichever*
WP	*wh*-pronoun, personal	*what, who, whom*
WP$	*wh*-pronoun, possessive	*whose, whosever*
WRB	*wh*-adverb	*where, when*
.	punctuation mark, sentence closer	*.;?**
,	punctuation mark, comma	*,*
:	punctuation mark, colon	*:*
(	contextual separator, left paren	*(*
)	contextual separator, right paren	*)*

UCREL CLAWS7 Tagset

APPGE	possessive pronoun, pre-nominal (e.g. my, your, our)
AT	article (e.g. the, no)
AT1	singular article (e.g. a, an, every)
BCL	before-clause marker (e.g. in order (that),in order (to))
CC	coordinating conjunction (e.g. and, or)
CCB	adversative coordinating conjunction (but)
CS	subordinating conjunction (e.g. if, because, unless, so, for)
CSA	as (as conjunction)
CSN	than (as conjunction)
CST	that (as conjunction)
CSW	whether (as conjunction)
DA	after-determiner or post-determiner capable of pronominal function (e.g. such, former, same)
DA1	singular after-determiner (e.g. little, much)
DA2	plural after-determiner (e.g. few, several, many)
DAR	comparative after-determiner (e.g. more, less, fewer)
DAT	superlative after-determiner (e.g. most, least, fewest)
DB	before determiner or pre-determiner capable of pronominal function (all, half)
DB2	plural before-determiner (both)
DD	determiner (capable of pronominal function) (e.g any, some)
DD1	singular determiner (e.g. this, that, another)
DD2	plural determiner (these,those)
DDQ	wh-determiner (which, what)
DDQGE	wh-determiner, genitive (whose)
DDQV	wh-ever determiner, (whichever, whatever)
EX	existential there
FO	formula
FU	unclassified word
FW	foreign word
GE	germanic genitive marker - (' or's)
IF	for (as preposition)
II	general preposition
IO	of (as preposition)
IW	with, without (as prepositions)
JJ	general adjective
JJR	general comparative adjective (e.g. older, better, stronger)
JJT	general superlative adjective (e.g. oldest, best, strongest)
JK	catenative adjective (able in be able to, willing in be willing to)
MC	cardinal number,neutral for number (two, three..)
MC1	singular cardinal number (one)
MC2	plural cardinal number (e.g. sixes, sevens)
MCGE	genitive cardinal number, neutral for number (two's, 100's)
MCMC	hyphenated number (40-50, 1770-1827)
MD	ordinal number (e.g. first, second, next, last)
MF	fraction,neutral for number (e.g. quarters, two-thirds)
ND1	singular noun of direction (e.g. north, southeast)
NN	common noun, neutral for number (e.g. sheep, cod, headquarters)
NN1	singular common noun (e.g. book, girl)
NN2	plural common noun (e.g. books, girls)
NNA	following noun of title (e.g. M.A.)
NNB	preceding noun of title (e.g. Mr., Prof.)
NNL1	singular locative noun (e.g. Island, Street)
NNL2	plural locative noun (e.g. Islands, Streets)
NNO	numeral noun, neutral for number (e.g. dozen, hundred)
NNO2	numeral noun, plural (e.g. hundreds, thousands)
NNT1	temporal noun, singular (e.g. day, week, year)
NNT2	temporal noun, plural (e.g. days, weeks, years)
NNU	unit of measurement, neutral for number (e.g. in, cc)
NNU1	singular unit of measurement (e.g. inch, centimetre)
NNU2	plural unit of measurement (e.g. ins., feet)
NP	proper noun, neutral for number (e.g. IBM, Andes)
NP1	singular proper noun (e.g. London, Jane, Frederick)
NP2	plural proper noun (e.g. Browns, Reagans, Koreas)
NPD1	singular weekday noun (e.g. Sunday)
NPD2	plural weekday noun (e.g. Sundays)
NPM1	singular month noun (e.g. October)
NPM2	plural month noun (e.g. Octobers)
PN	indefinite pronoun, neutral for number (none)
PN1	indefinite pronoun, singular (e.g. anyone, everything, nobody, one)
PNQO	objective wh-pronoun (whom)
PNQS	subjective wh-pronoun (who)
PNQV	wh-ever pronoun (whoever)
PNX1	reflexive indefinite pronoun (oneself)
PPGE	nominal possessive personal pronoun (e.g. mine, yours)

Lexical Analysis 4: Part-of-Speech (POS) Tagging

- Tagsets: Korean

한글 형태소 품사 (Part Of Speech, POS) 태그표

한글 형태소의 품사를 **체언, 용언, 관형사, 부사, 감탄사, 조사, 어미, 접사, 어근, 부호, 한글 이외**'와 같이 나누고 각 세부 품사를 구분한다.

대분류	세종 품사 태그		심광섭 품사 태그		KKMA 단일 태그 V 1.0					
	태그	설명	Class	설명	묶음 1	묶음 2	태그	설명	확률태그	저장사전
체언	NNG	일반 명사	NN	명사	N	NN	NNG	보통 명사	NNA	noun.dic
	NNP	고유 명사					NNP	고유 명사		
	NNB	의존 명사	NX	의존 명사			NNB	일반 의존 명사	NNB	simple.dic
			UM	단위 명사			NNM	단위 의존 명사		
	NR	수사	NU	수사		NR	NR	수사	NR	
	NP	대명사	NP	대명사		NP	NP	대명사	NP	
용언	VV	동사	VV	동사	V	VV	VV	동사	VV	verb.dic
	VA	형용사	AJ	형용사		VA	VA	형용사	VA	
	VX	보조 용언	VX	보조 동사		VX	VXV	보조 동사	VX	
			AX	보조 형용사			VXA	보조 형용사		
	VCP	긍정 지정사	CP	서술격 조사 '이다'		VC	VCP	긍정 지정사, 서술격 조사 '이다'	VCP	raw.dic
	VCN	부정 지정사					VCN	부정 지정사, 형용사 '아니다'	VCN	
관형사	MM	관형사	DT	일반 관형사	M	MD	MDT	일반 관형사	MD	simple.dic
			DN	수 관형사			MDN	수 관형사		
부사	MAG	일반 부사	AD	부사		MA	MAG	일반 부사	MAG	
	MAJ	접속 부사					MAC	접속 부사	MAC	
감탄사	IC	감탄사	EX	감탄사	I	IC	IC	감탄사	IC	
조사	JKS	주격 조사	JO	조사	J	JK	JKS	주격 조사	JKS	
	JKC	보격 조사					JKC	보격 조사	JKC	
	JKG	관형격 조사					JKG	관형격 조사	JKG	
	JKO	목적격 조사					JKO	목적격 조사	JKO	
	JKB	부사격 조사					JKM	부사격 조사	JKM	
	JKV	호격 조사					JKI	호격 조사	JKI	
	JKQ	인용격 조사					JKQ	인용격 조사	JKQ	
	JX	보조사				JX	JX	보조사	JX	
	JC	접속 조사				JC	JC	접속 조사	JC	
선어말 어미	EP	선어말 어미	EP	선어말 어미		EP	EPH	존칭 선어말 어미	EP	
							EPT	시제 선어말 어미		
							EPP	공손 선어말 어미		raw.dic

어말 어미	EF	종결 어미	EM	어말 어미	E	EF	EFN	평서형 종결 어미	EF	
							EFQ	의문형 종결 어미		
							EFO	명령형 종결 어미		
							EFA	청유형 종결 어미		
							EFI	감탄형 종결 어미		
							EFR	존칭형 종결 어미		
	EC	연결 어미				EC	ECE	대등 연결 어미	EC	
							ECD	의존적 연결 어미		
							ECS	보조적 연결 어미		
	ETN	명사형 전성 어미				ET	ETN	명사형 전성 어미	ETN	
	ETM	관형형 전성 어미					ETD	관형형 전성 어미	ETD	
접두사	XPN	체언 접두사	PF	접두사	X	XP	XPN	체언 접두사	XPN	simple.dic
							XPV	용언 접두사	XPV	
접미사	XSN	명사 파생 접미사	SN	명사화 접미사		XS	XSN	명사 파생 접미사	XSN	
	XSV	동사 파생 접미사	SV	동사화 접미사			XSV	동사 파생 접미사	XSV	
	XSA	형용사 파생 접미사	SJ	형용사화 접미사			XSA	형용사 파생 접미사	XSA	
			SA	부사화 접미사			~~XSM~~	~~부사 파생 접미사~~	~~XSM~~	
			SF	기타 접미사			~~XSO~~	~~기타 접미사~~	~~XSO~~	
어근	XR	어근	XR			XR	XR	어근	XR	
부호	SF	마침표물음표,느낌표	SY	부호 외래어	S	SF	SF	마침표물음표,느낌표	SF	Symbol class
	SP	쉼표,가운뎃점,콜론,빗금				SP	SP	쉼표,가운뎃점,콜론,빗금	SP	
	SS	따옴표,괄호표,줄표				SS	SS	따옴표,괄호표,줄표	SS	
	SE	줄임표				SE	SE	줄임표	SE	
	SO	붙임표(물결,숨김,빠짐)				SO	SO	붙임표(물결,숨김,빠짐)	SO	
	SW	기타기호 (논리수학기호,화폐기호)				SW	SW	기타기호 (논리수학기호,화폐기호)	SW	
분석 불능	NF	명사추정범주	NR	미등록어	U	UN	UN	명사추정범주	NNA	N/A
	NV	용언추정범주				UV	~~UV~~	~~용언추정범주~~	~~N/A~~	
	NA	분석불능범주				UE	~~UE~~	~~분석불능범주~~	~~N/A~~	
한글 이외	SL	외국어			O	OL	OL	외국어	NNA	
	SH	한자				OH	OH	한자	NNA	
	SN	숫자				ON	ON	숫자	NR	

Lexical Analysis 4: Part-of-Speech (POS) Tagging

Witte (2016)

- POS Tagging Algorithms

Fundamentals

POS-Tagging generally requires:

Training phase where a manually annotated corpus is processed by a machine learning algorithm; and a

Tagging algorithm that processes texts using learned parameters.

Performance is generally good (around 96%) when staying in the same domain.

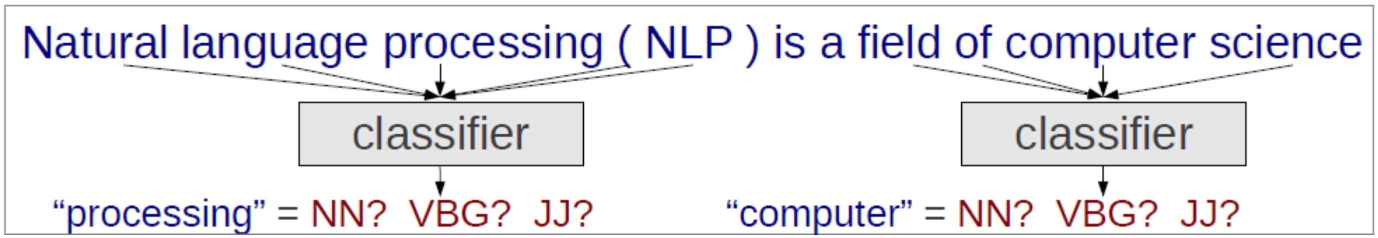
Algorithms used in POS-Tagging

There is a multitude of approaches, commonly used are:

- Decision Trees
- Hidden Markov Models (HMMs)
- Support Vector Machines (SVM)
- Transformation-based Taggers (e.g., the Brill tagger)

Lexical Analysis 4: Part-of-Speech (POS) Tagging

- POS Tagging Algorithms
  - ✓ Pointwise prediction: predict each word individually with a classifier (e.g. Maximum Entropy Model, SVM)



  - ✓ Probabilistic models
    - Generative sequence models: Find the most probable tag sequence given the sentence (Hidden Markov Model; HMM)
    - Discriminative sequence models: Predict whole sequence with a classifier (Conditional Random Field; CRF)
  - ✓ Neural network-based models

Lexical Analysis 4: Part-of-Speech (POS) Tagging

- Pointwise Prediction: Maximum Entropy Model
  - ✓ Encode features for tag prediction
    - Information about word/context: suffix, prefix, neighborhood word information
    - eg: $f_i(w_j, t_j)$ = 1 if suffix(w_j) = "ing" & t_j = VBG, 0 otherwise
  - ✓ Tagging Model

$$p(t|C) = \frac{1}{Z(C)} \exp\Big(\sum_{i=1}^{n} \lambda_i f_i(C,t) \Big) \qquad p(t_1, \ldots, t_n | w_1, \ldots, w_n) \approx \prod_{i=1}^{n} p(t_i | w_i)$$

    - f_i is a feature
    - λi is a weight (large value implies informative features)
    - Z(C) is a normalization constant ensuring a proper probability distribution
    - Makes no independence assumption about the features

Lexical Analysis 4: Part-of-Speech (POS) Tagging

- Pointwise Prediction: Maximum Entropy Model
  - ✓ An example

```
# POS Tagging with MaxEnt
install.packages("openNLP")
library(openNLP)

s1 <- paste(c("Pierre Vinken, 61 years old, will join the board as a ",
              "nonexecutive director Nov. 29.\n",
              "Mr. Vinken is chairman of Elsevier N.V., ",
              "the Dutch publishing group."),
            collapse = "")
s1 <- as.String(s1)
```

```
> s1
Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.
Mr. Vinken is chairman of Elsevier N.V., the Dutch publishing group.
```

Lexical Analysis 4: Part-of-Speech (POS) Tagging

- Pointwise Prediction: Maximum Entropy Model

  ✓ An example

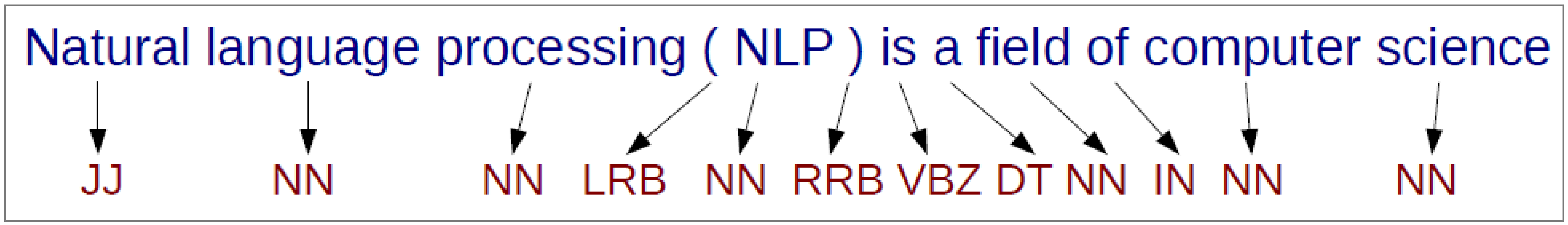
```r
## Need sentence and word token annotations.
s2 <- annotate(s1, list(Maxent_Sent_Token_Annotator(), Maxent_Word_Token_Annotator()))

## POS tag probabilities as (additional) features.
s3 <- annotate(s1, Maxent_POS_Tag_Annotator(probs = TRUE), s2)
```

```
> s3
 id type     start end features
  1 sentence     1  84 constituents=<<integer,18>>
  2 sentence    86 153 constituents=<<integer,13>>
  3 word         1   6 POS=NNP, POS_prob=0.9476405
  4 word         8  13 POS=NNP, POS_prob=0.9692841
  5 word        14  14 POS=,, POS_prob=0.9884445
  6 word        16  17 POS=CD, POS_prob=0.9926943
  7 word        19  23 POS=NNS, POS_prob=0.9893489
  8 word        25  27 POS=JJ, POS_prob=0.9693832
  9 word        28  28 POS=,, POS_prob=0.9873552
 10 word        30  33 POS=MD, POS_prob=0.9460105
 11 word        35  38 POS=VB, POS_prob=0.9865564
 12 word        40  42 POS=DT, POS_prob=0.9692801
 13 word        44  48 POS=NN, POS_prob=0.9928681
 14 word        50  51 POS=IN, POS_prob=0.9592474
 15 word        53  53 POS=DT, POS_prob=0.9890297
 16 word        55  66 POS=JJ, POS_prob=0.7213763
 17 word        68  75 POS=NN, POS_prob=0.987327
 18 word        77  80 POS=NNP, POS_prob=0.9581523
 19 word        82  83 POS=CD, POS_prob=0.9502215
 20 word        84  84 POS=., POS_prob=0.9943433
 21 word        86  88 POS=NNP, POS_prob=0.9762001
 22 word        90  95 POS=NNP, POS_prob=0.9904051
 23 word        97  98 POS=VBZ, POS_prob=0.9820713
 24 word       100 107 POS=NN, POS_prob=0.8300819
 25 word       109 110 POS=IN, POS_prob=0.9838273
 26 word       112 119 POS=NNP, POS_prob=0.9231359
 27 word       121 124 POS=NNP, POS_prob=0.9969889
 28 word       125 125 POS=,, POS_prob=0.9762171
 29 word       127 129 POS=DT, POS_prob=0.9811851
 30 word       131 135 POS=JJ, POS_prob=0.8021723
 31 word       137 146 POS=NN, POS_prob=0.9669352
 32 word       148 152 POS=NN, POS_prob=0.9940887
 33 word       153 153 POS=., POS_prob=0.9898899
```

Lexical Analysis 4: Part-of-Speech (POS) Tagging

- Probabilistic Model for POS Tagging
  - ✓ Find the most probable tag sequence given the sentence



$$\underset{Y}{\operatorname{argmax}}\, P(Y|X)$$

Lexical Analysis 4: Part-of-Speech (POS) Tagging

- Generative Sequence Model
  - ✓ Decompose probability using Baye's Rule

$$\underset{Y}{\operatorname{argmax}} P(Y|X) = \underset{Y}{\operatorname{argmax}} \frac{P(X|Y)P(Y)}{P(X)}$$
$$= \underset{Y}{\operatorname{argmax}} P(X|Y)P(Y)$$

Model of word/POS interactions
"natural" is probably a JJ

Model of POS/POS interactions
NN comes after DET

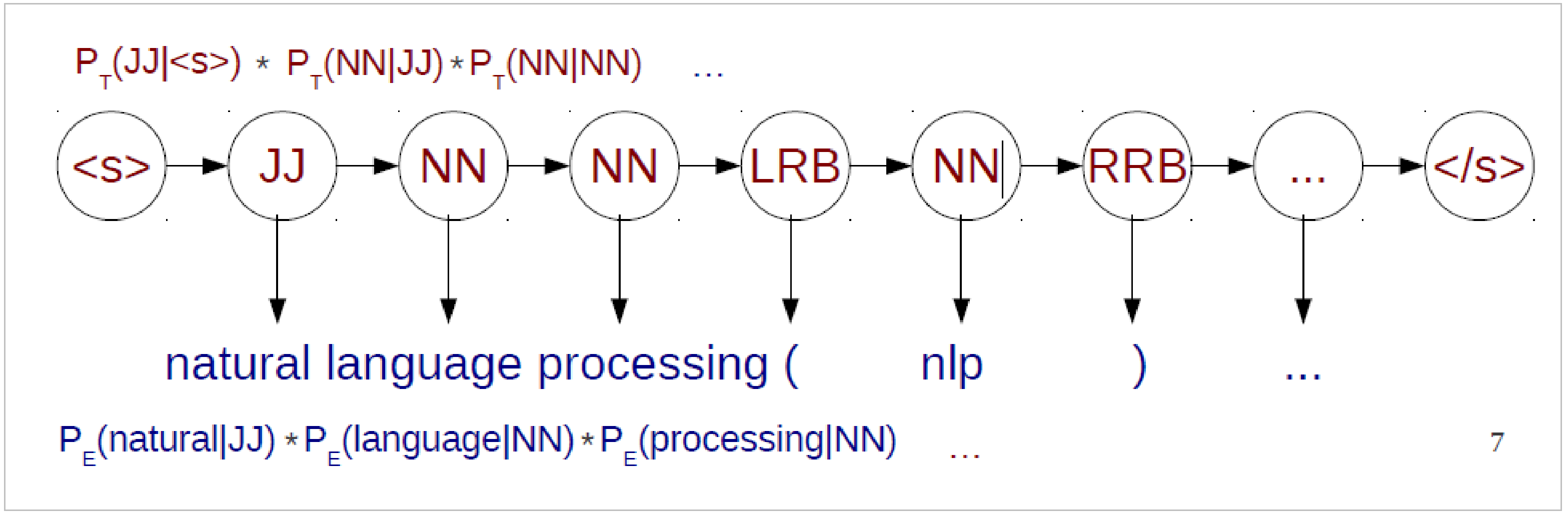
Lexical Analysis 4: Part-of-Speech (POS) Tagging

- Generative Sequence Model: Hidden Markov Model
  - ✓ POS → POS transition probabilities

$$P(Y) \approx \prod_{i=1}^{I+1} P_T(y_i|y_{i-1})$$

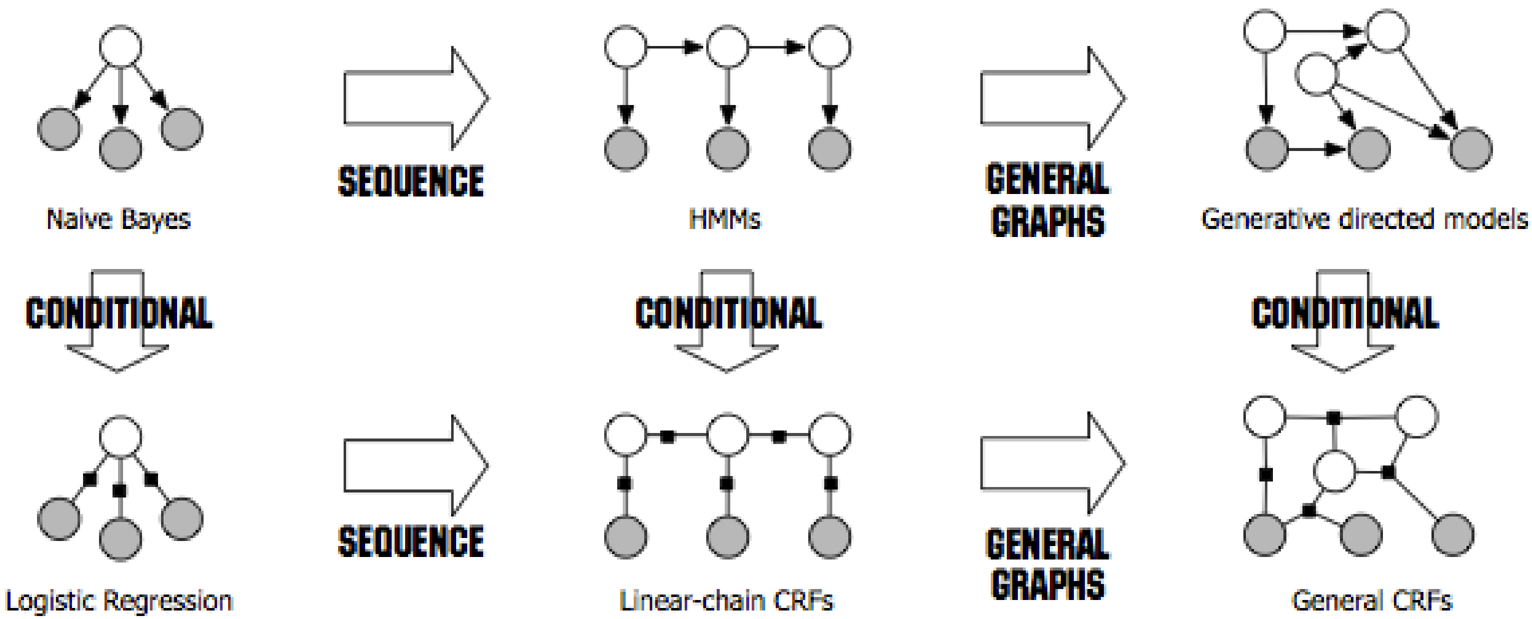
  - ✓ POS → Word emission probabilities

$$P(X|Y) \approx \prod_{1}^{I} P_E(x_i|y_i)$$



Lexical Analysis 4: Part-of-Speech (POS) Tagging

- Discriminative Sequence Model: Conditional Random Field (CRF)
  - ✓ Relieve that constraint that a tag is generated by the previous tag sequence
  - ✓ Predict the whole tag set at the same time, not sequentially

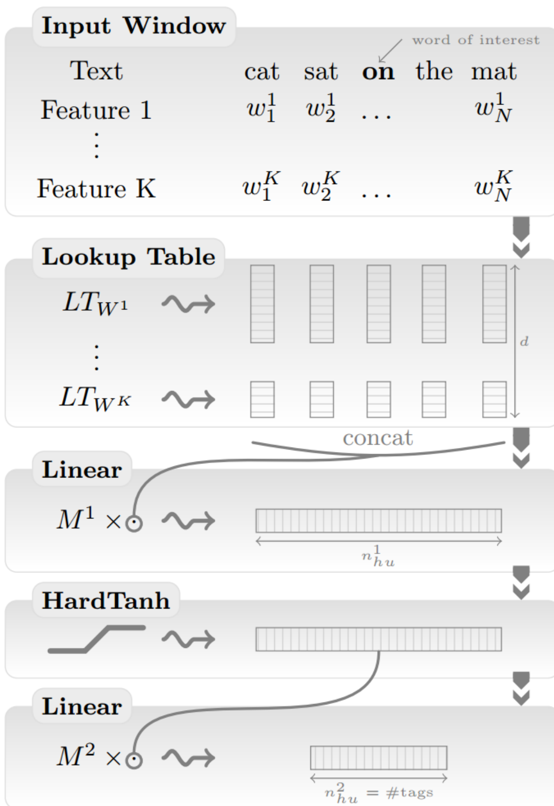


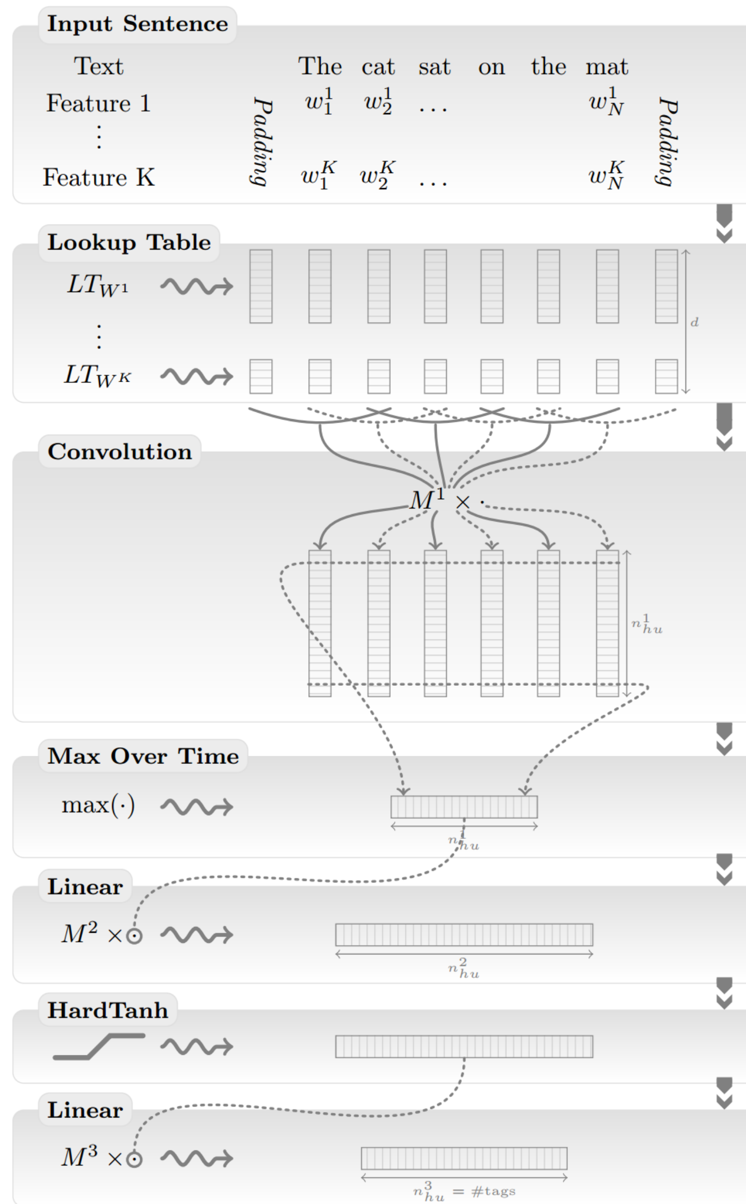
http://people.cs.umass.edu/~mccallum/papers/crf-tutorial.pdf

Lexical Analysis 4: Part-of-Speech (POS) Tagging

Collobert et al. (2011)

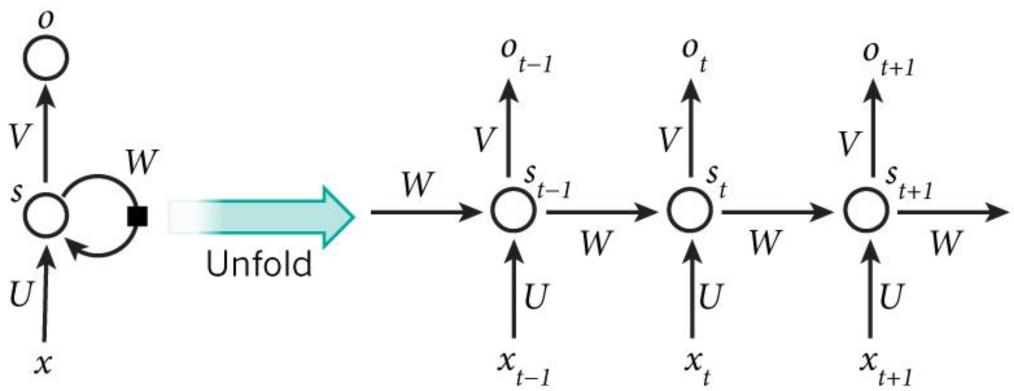
- Neural Network-based Models
  - ✓ Window-based vs. sentence-based



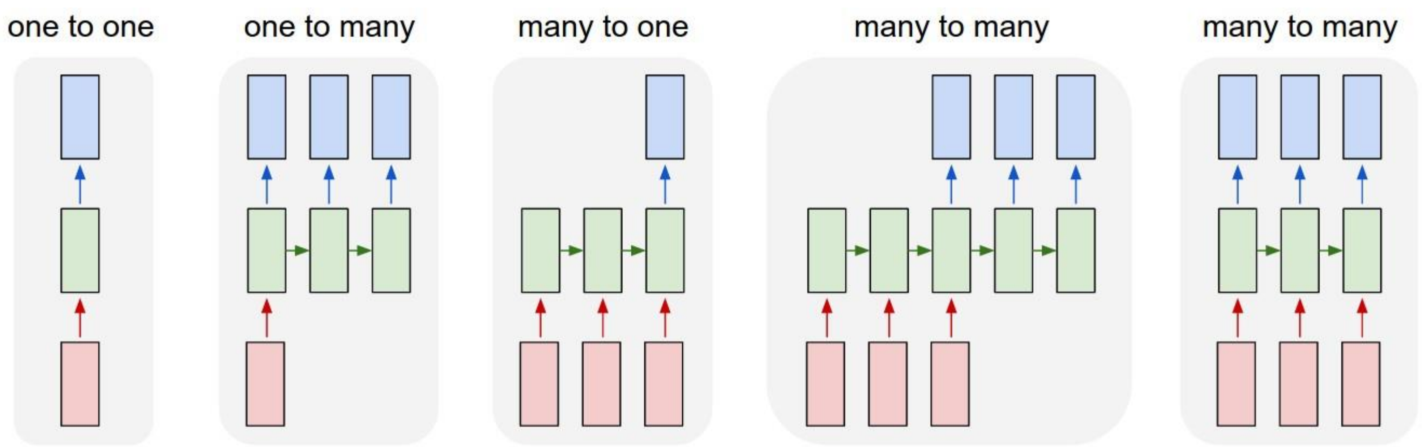


Lexical Analysis 4: Part-of-Speech (POS) Tagging

- Neural network-based models
  - ✓ Recurrent neural networks: have a feedback loop within the hidden layer

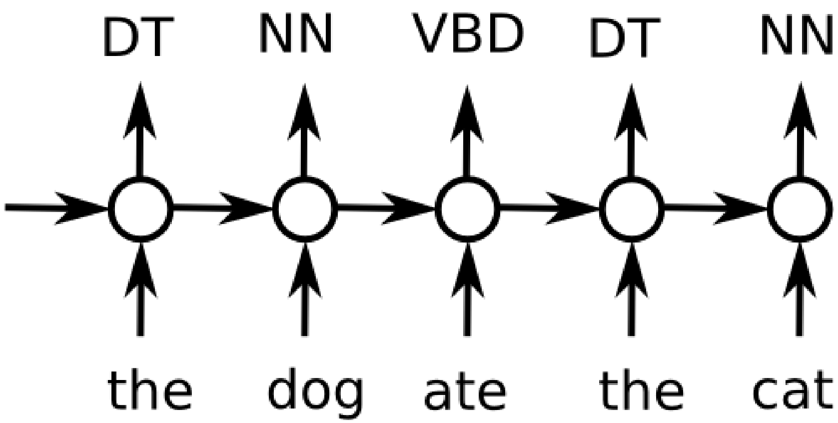


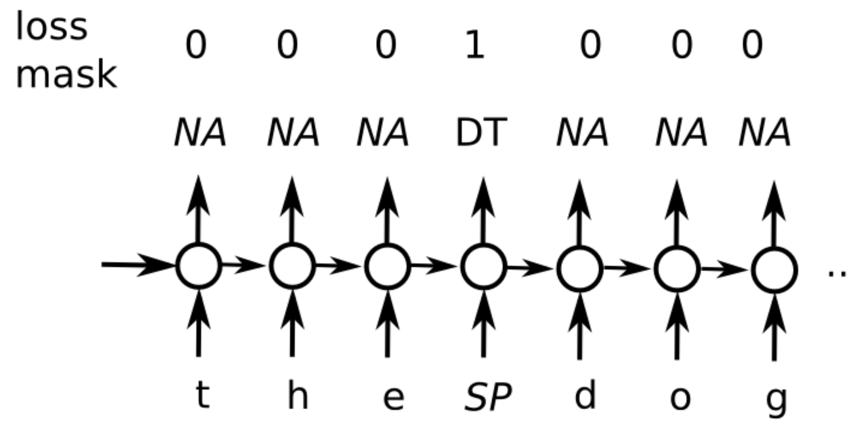
  - ✓ Input-Output mapping of RNNs



Lexical Analysis 4: Part-of-Speech (POS) Tagging

- Neural network-based models: Recurrent neural networks

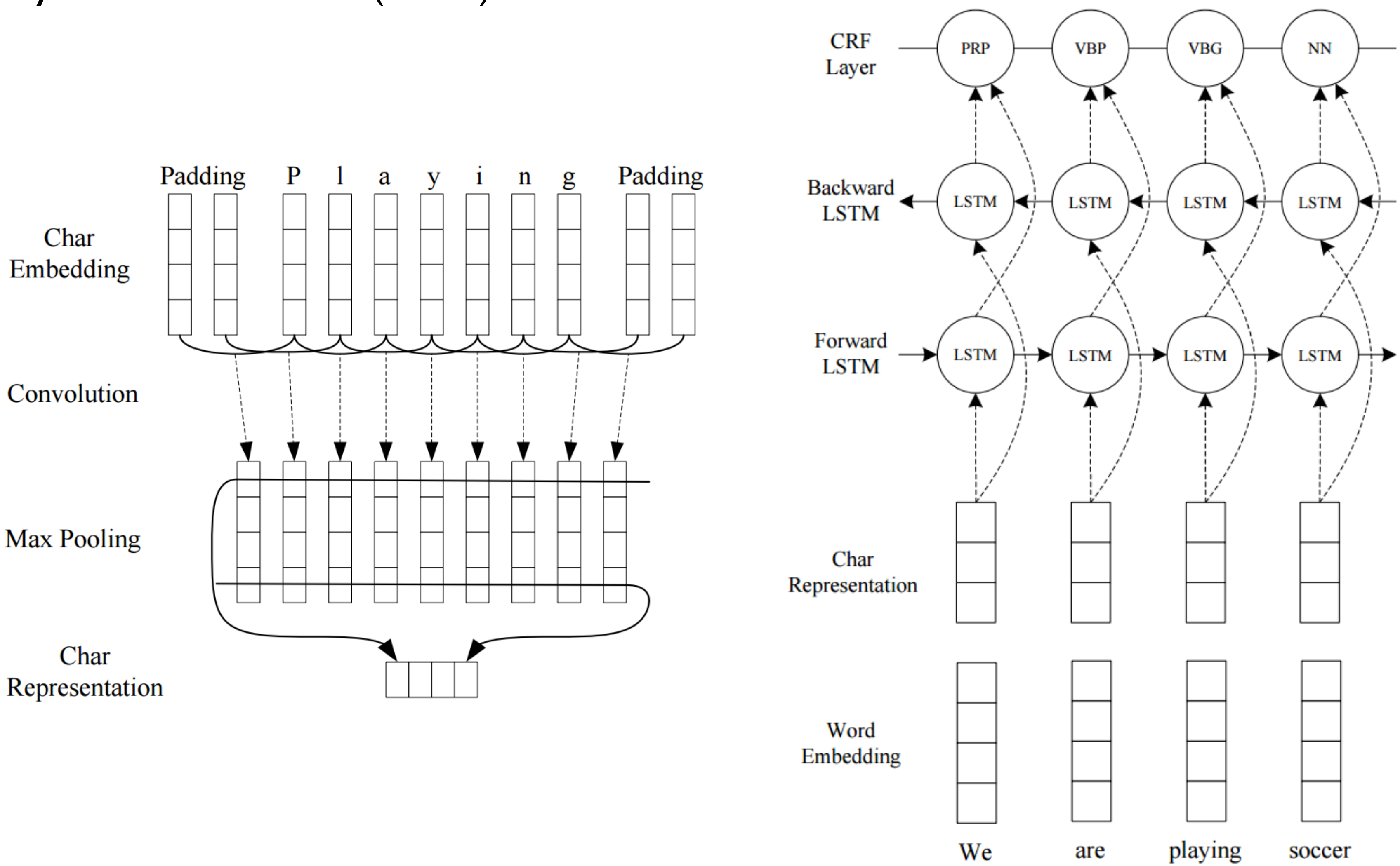




Lexical Analysis 4: Part-of-Speech (POS) Tagging

Ma and Hovy (2016)

- Hybrid model: LSTM(RNN) + ConvNet + CRF



Lexical Analysis 5: Named Entity Recognition

- Named Entity Recognition: NER
  - ✓ a subtask of information extraction that seeks to locate and classify elements in text into pre-defined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.

http://eric-yuan.me/ner_1/

Lexical Analysis 5: Named Entity Recognition

Approaches for NER: Dictionary/Rule-based

- List lookup: systems that recognizes only entities stored in its lists
  - ✓ Advantages: simple, fast, language independent, easy to retarget.
  - ✓ Disadvantages: collection and maintenance of list cannot deal with name variants and cannot resolve ambiguity

- Shallow Parsing Approach
  - ✓ Internal evidence – names often have internal structure. These components can be either stored or guessed.
    - Location: Cap Word + {Street, Boulevard, Avenue, Crescent, Road}
    - e.g.: Wall Street

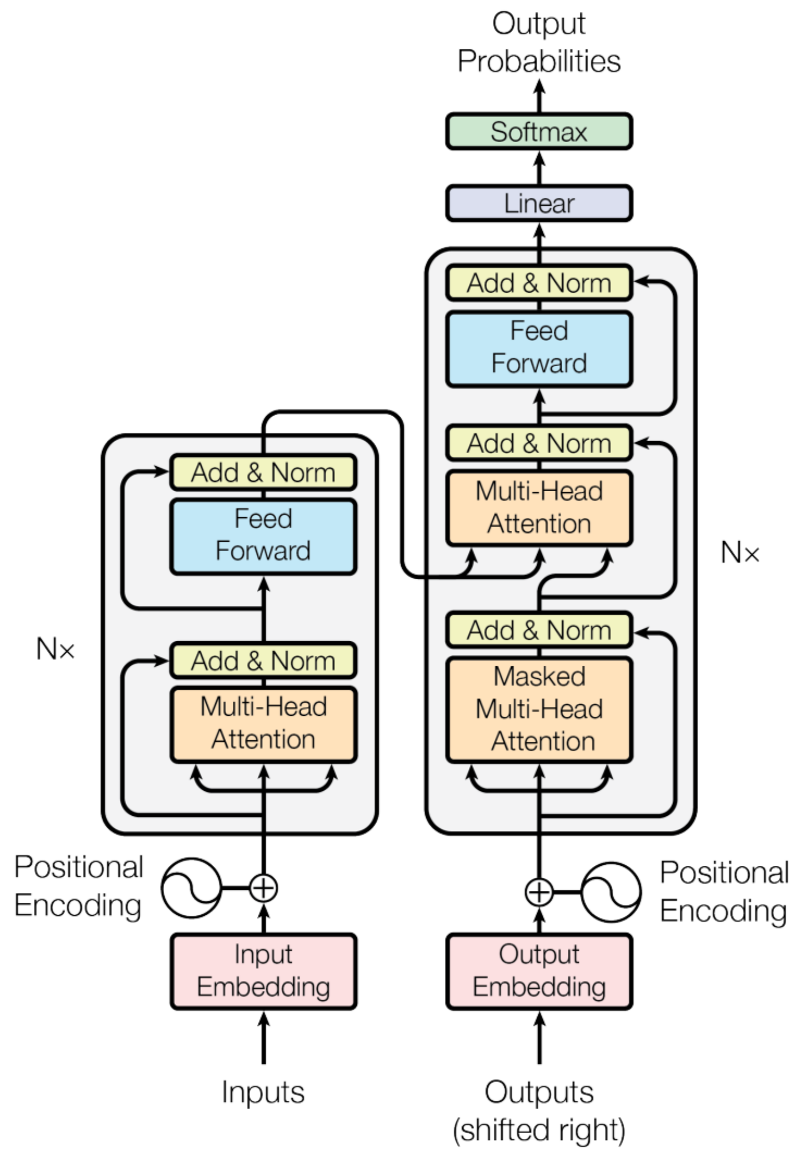
Lexical Analysis 5: Named Entity Recognition

<u>Approaches for NER: Model-based</u>

- MITIE
  - ✓ An open sourced information extraction tool developed by MIT NLP lab.
  - ✓ Available for English and Spanish
  - ✓ Available for C++, Java, R, and Python
- CRF++
  - ✓ NER based on conditional random fields
  - ✓ Supports multi-language models
- Convolutional neural networks
  - ✓ 1-of-M coding, Word2Vec, N-Grams can be used as encoding methods

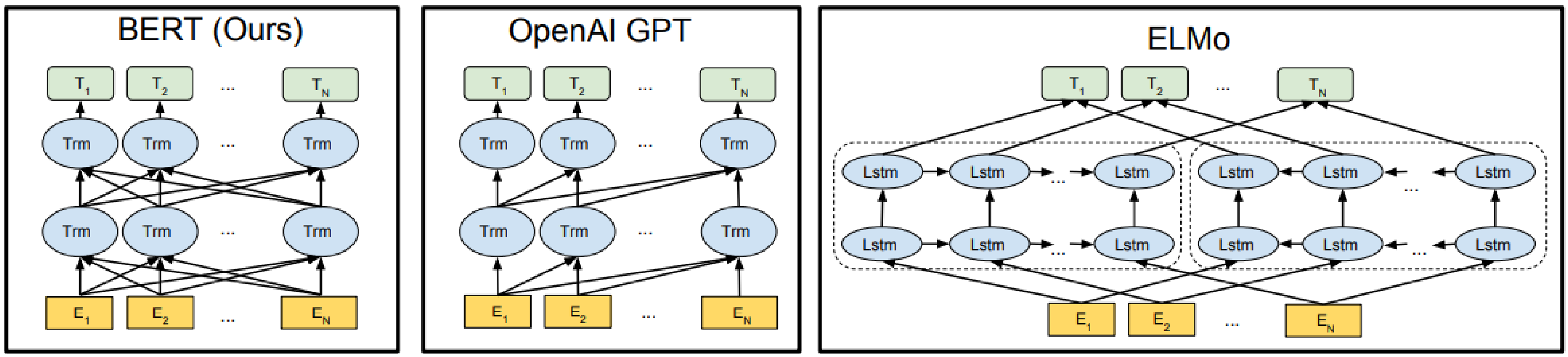
BERT for Multi NLP Tasks

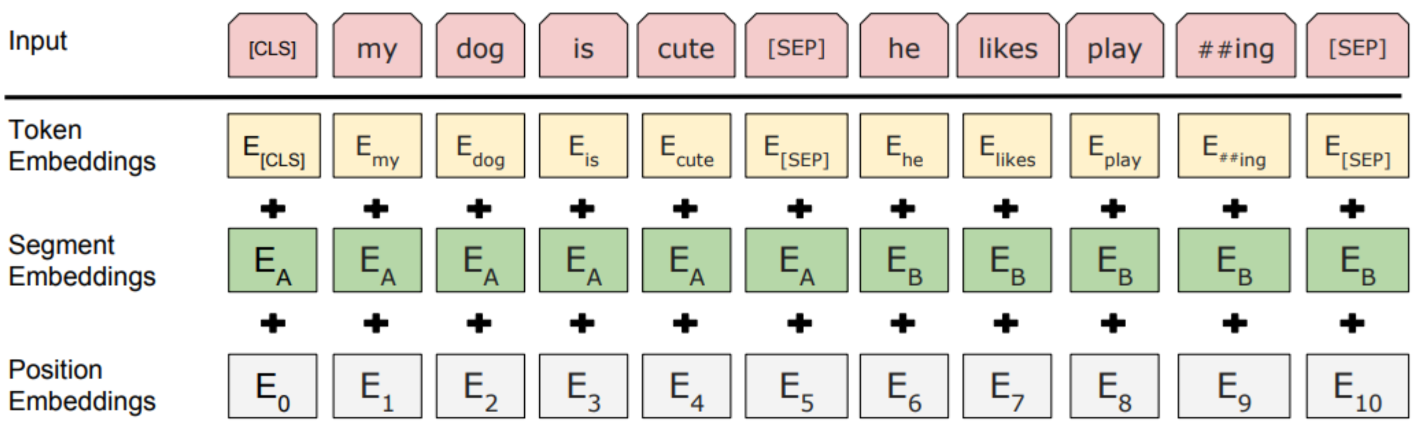
- Google Transformer
  - ✓ Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*(pp. 5998-6008).
  - ✓ Excellent blog post explaining Transformer
    - http://jalammar.github.io/illustrated-transformer/



BERT for Multi NLP Tasks

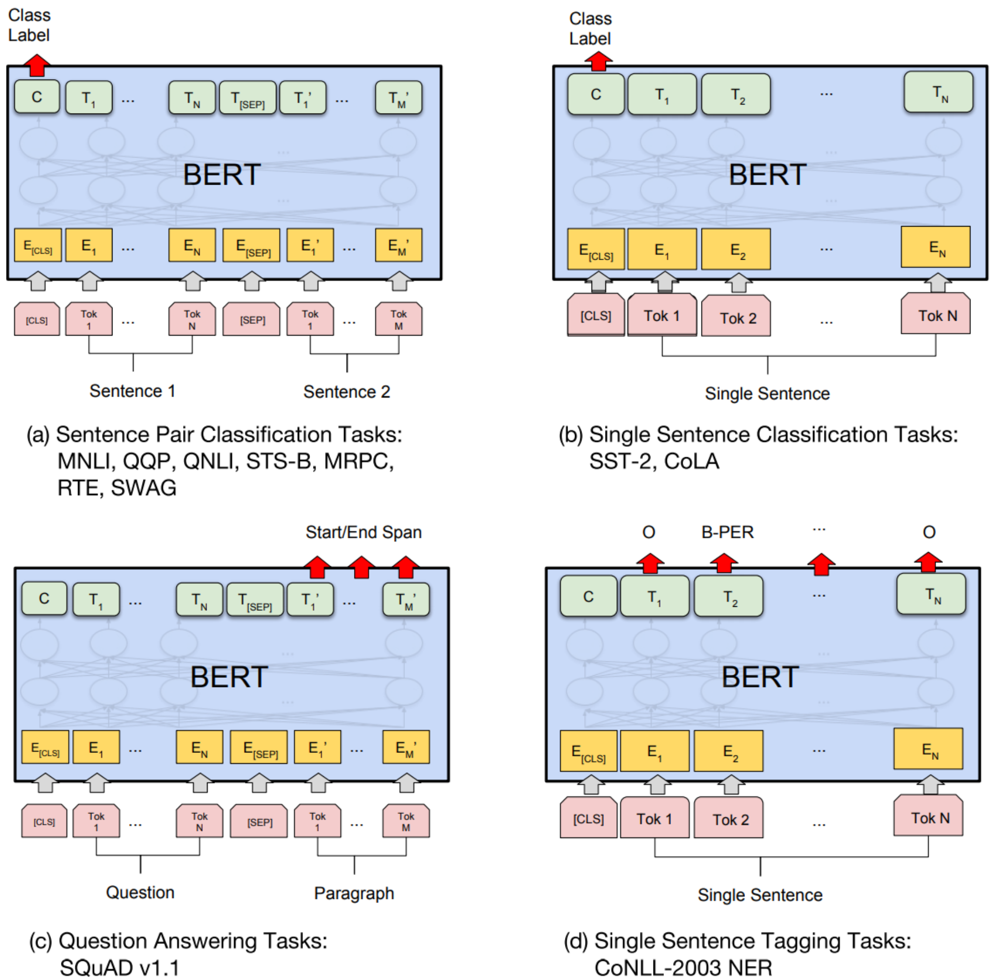
- BERT
  - ✓ Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.





BERT for Multi NLP Tasks

- BERT
  - ✓ Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

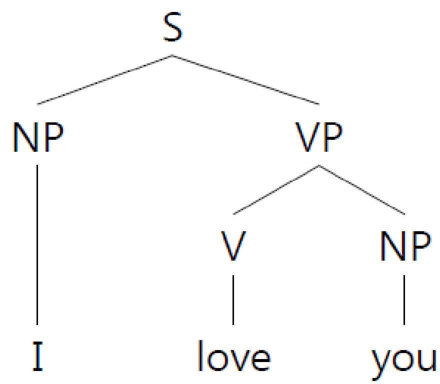
(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

AGENDA

Syntax Analysis

- Syntax Analysis
  - ✓ Process of analyzing a string of symbols conforming to the rules of a formal grammar

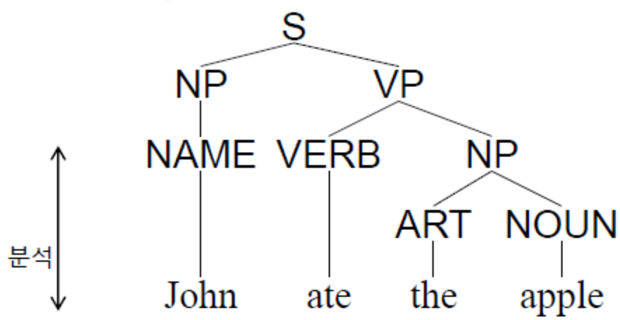


- Parser
  - ✓ An algorithm that computes a structure for an input string given a grammar
  - ✓ All parsers have two fundamental properties
    - Directionality: the sequence in which the structures are constructed (e.g., top-down or bottom-up)
    - Search strategy: the order in which the search space of possible analysis explored (e.g., depth-first, breadth-first)

Syntax Analysis

- Parsing Representation
  - ✓ Tree vs List

Tree Representation



List Representation

(S (NP (NAME John))
(VP (VERB ate)
(NP (ART the)
(NOUN apple))))

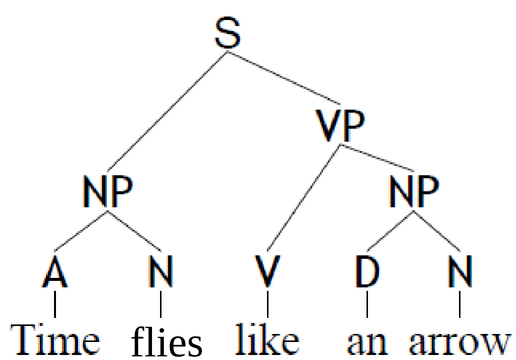
  - ✓ Meaning
    - S (Sentence) consists of NP (Noun Phrase) and VP (Verb Phrase)
    - NP consists of Name (John)
    - VP consists of VERB (ate) and the other NP
    - NP consists of ART (the) and Noun (apple)

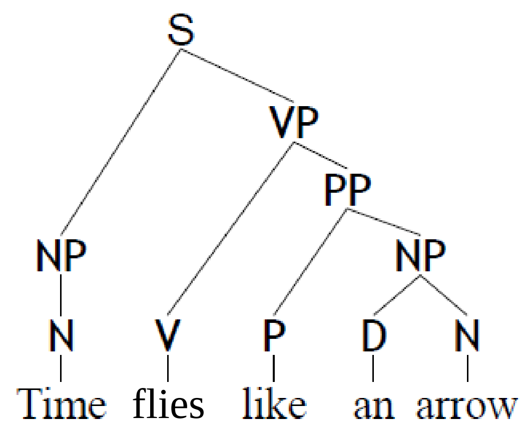
Syntax Analysis

- Not a single parsing tree due to language ambiguity
- Lexical ambiguity
  - ✓ One word can be used for multiple parts of speech
  - ✓ Lexical ambiguity causes structural ambiguity

G :
S $\rightarrow$ NP VP
NP $\rightarrow$ D N | A N | N
VP $\rightarrow$ V | VP NP | VP PP
PP $\rightarrow$ P NP

Input Sentence :
Time flies like an arrow



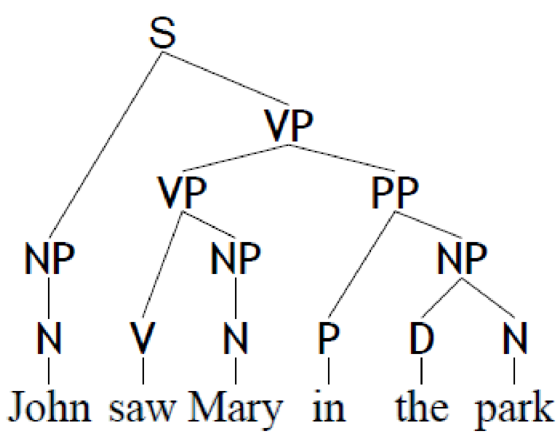


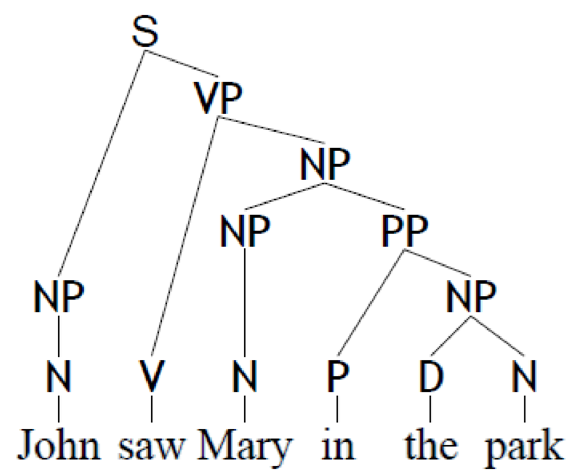
Syntax Analysis

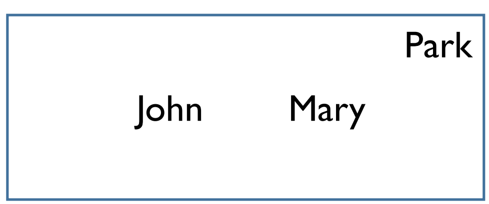
- Structural Ambiguity
  - ✓ One sentence can be understood in different ways

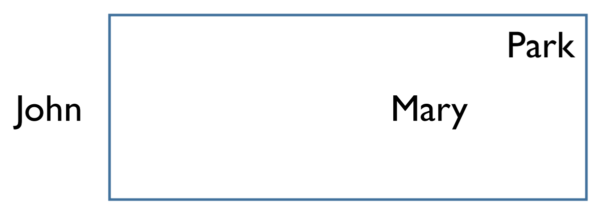
G : S $\rightarrow$ NP VP
NP $\rightarrow$ N | D N | NP PP
VP $\rightarrow$ V NP | VP PP
PP $\rightarrow$ P NP

Input Sentence :
John saw Mary in the park.









AGENDA

Language Modeling

- Probabilistic Language Model
  - ✓ Assign a probability to a sentence (not POS tags, but the sentence itself)
- Applications
  - ✓ Machine Translation
    - P(**high** wind tonight) > P(**large** wind tonight)
  - ✓ Spell correction
    - The office is about fifteen **minuets** from my house
    - P(about fifteen **minutes** from) > P(about fifteen **minuets** from)
  - ✓ Speech recognition
    - P(I saw a van) >> P(eyes awe of an)
  - ✓ Summarization, question-answering, etc.

Language Modeling

Jurafsky, Language Modeling

- Probabilistic Language Modeling
  - ✓ Compute the probability of a sentence or sequence of words

$$P(W) = P(w_1, w_2, w_3, ..., w_n)$$

  - ✓ Related task: probability of an upcoming word

$$P(w_5|w_1, w_2, w_3, w_4)$$

    - ex) I love you more than I can ______. (swim? say?)

- How to compute P(W)
  - ✓ What is P(its, water, is, so, transparent, that)?
  - ✓ Chain Rules of Probability:

$$P(w_1, w_2, w_3, ..., w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)...P(w_n|w_1, ..., w_{n-1})$$

$$P(\text{its water is so trasparent”}) = P(\text{its}) \times P(\text{water}|\text{its}) \times P(\text{is}|\text{its water})$$
$$\times P(\text{so}|\text{its water is}) \times P(\text{transparent}|\text{its water is so})$$

Language Modeling

- Markov Assumption
  - ✓ Consider only k previous words when estimating the conditional probability

$$P(w_1 w_2 \ldots w_n) \approx \prod_i P(w_i | w_{i-k} \ldots w_{i-1}) \qquad P(w_i | w_1 w_2 \ldots w_{i-1}) \approx \prod_i P(w_i | w_{i-k} \ldots w_{i-1})$$

  - ✓ Simplest case: Unigram model

$$P(w_1 w_2 \ldots w_n) \approx \prod_i P(w_i)$$

  - ✓ An example of automatically generated sentences from a unigram model

```
fifth, an, of, futures, the, an, incorporated, a,
a, the, inflation, most, dollars, quarter, in, is,
mass

thrift, did, eighty, said, hard, 'm, july, bullish

that, or, limited, the
```

Language Modeling

- Bigram model
  - ✓ Condition on the previous word

$$P(w_i|w_1 w_2 \ldots w_{i-1}) \approx P(w_i|w_{i-1})$$

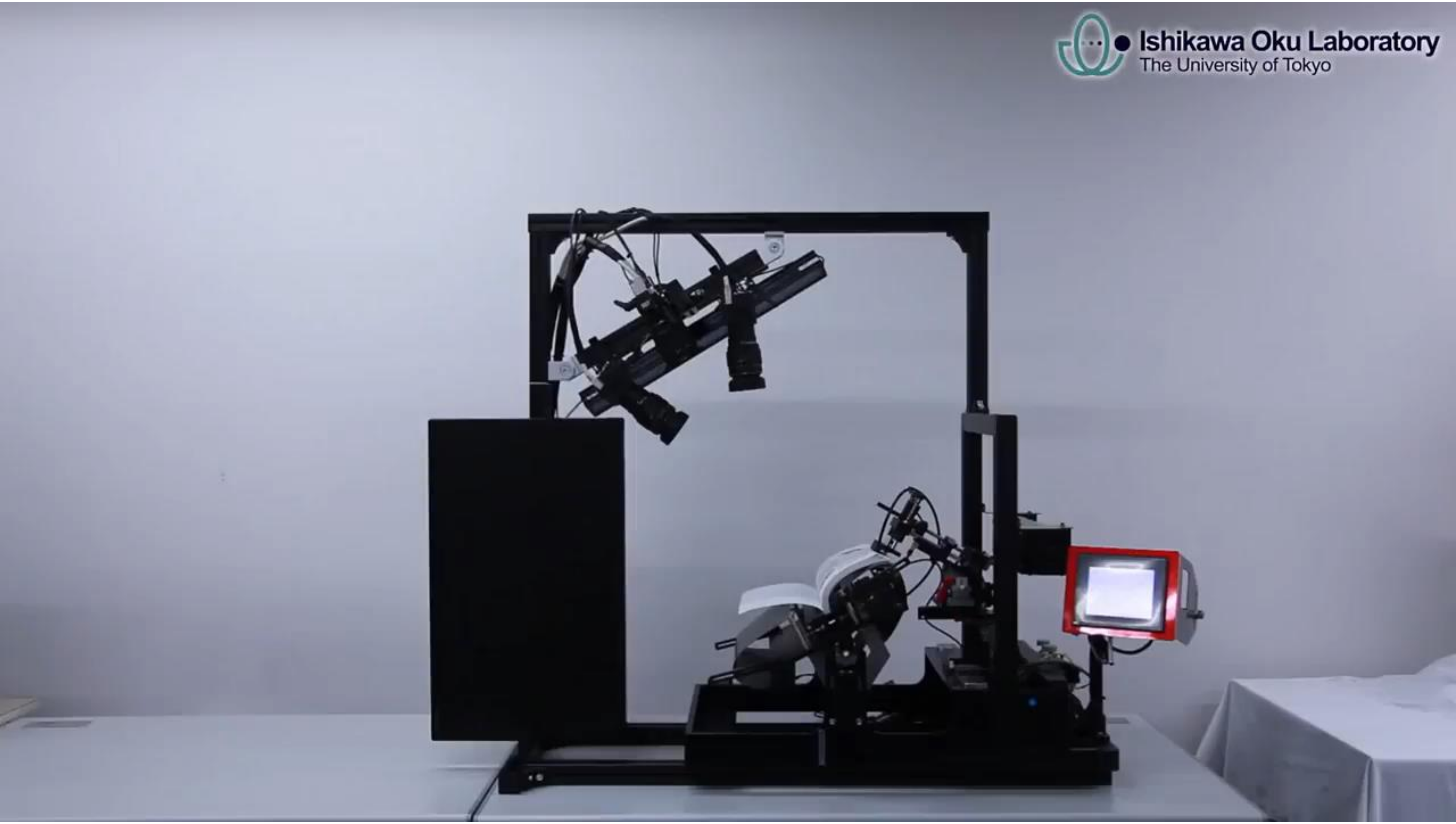
```
texaco, rose, one, in, this, issue, is, pursuing, growth, in,
a, boiler, house, said, mr., gurria, mexico, 's, motion,
control, proposal, without, permission, from, five, hundred,
fifty, five, yen

outside, new, car, parking, lot, of, the, agreement, reached

this, would, be, a, record, november
```

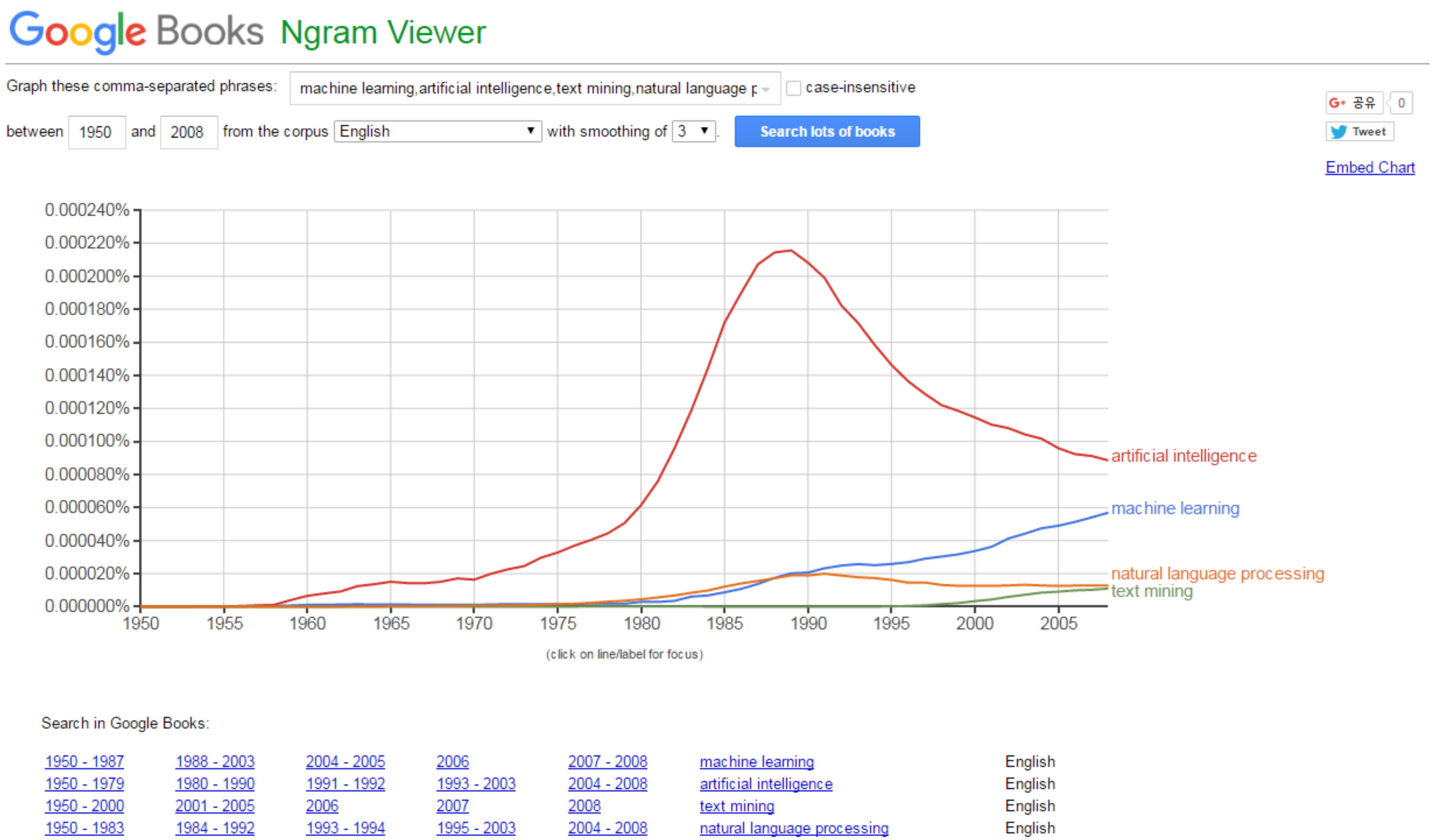
- N-gram models
  - ✓ Can extend to trigrams, 4-grams, 5-grams
    - In sufficient model of language because language has long-distance dependencies
    - "The computer when I had just put into the machine room on the fifth floor crashed."
  - ✓ We can often get away with N-gram models

Language Modeling



Language Modeling

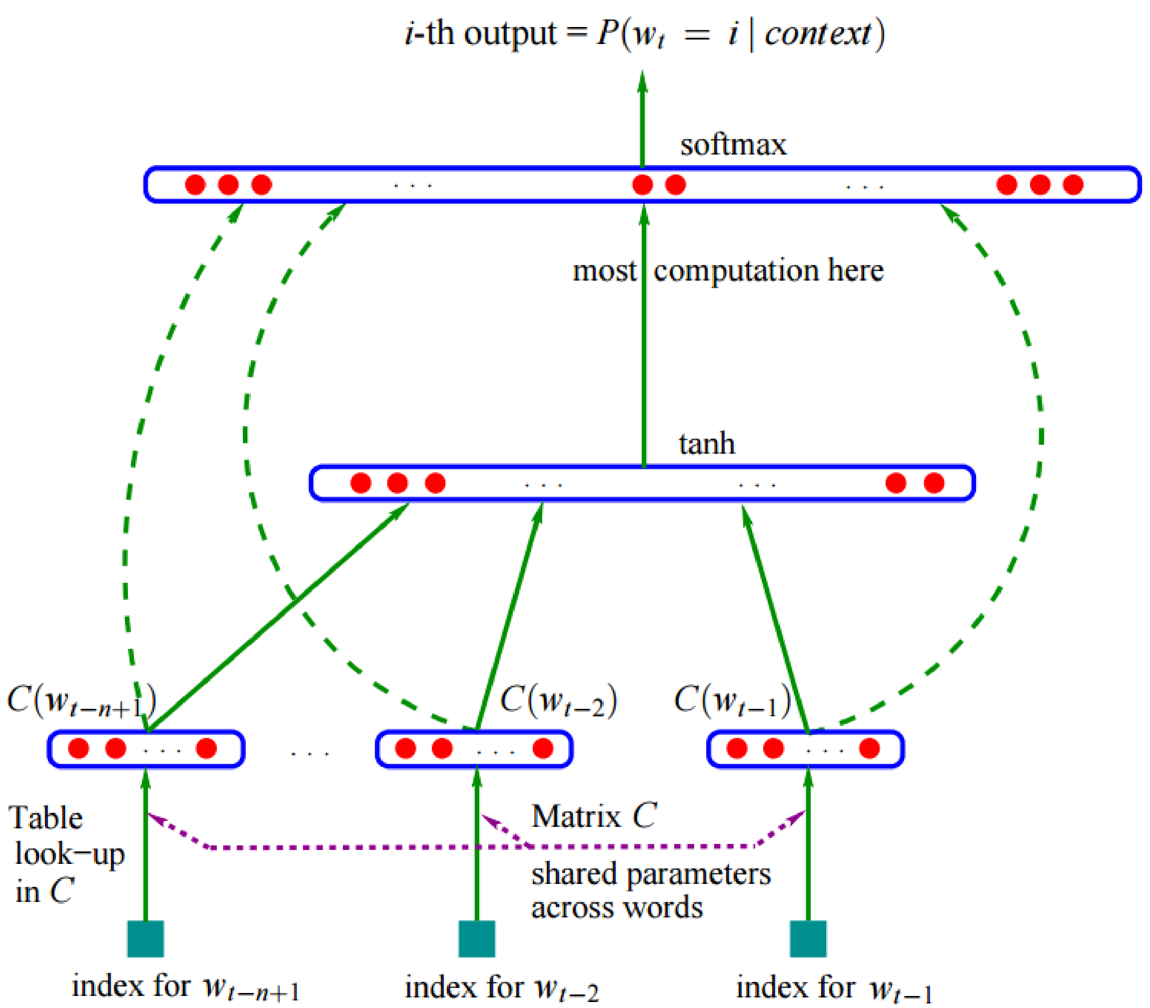
- Google Books N-Gram
  - ✓ 1,024 billion words & 1.1 billion 5-grams that appeared at least 40 times (2006)



Language Modeling

Bengio et al. (2003)

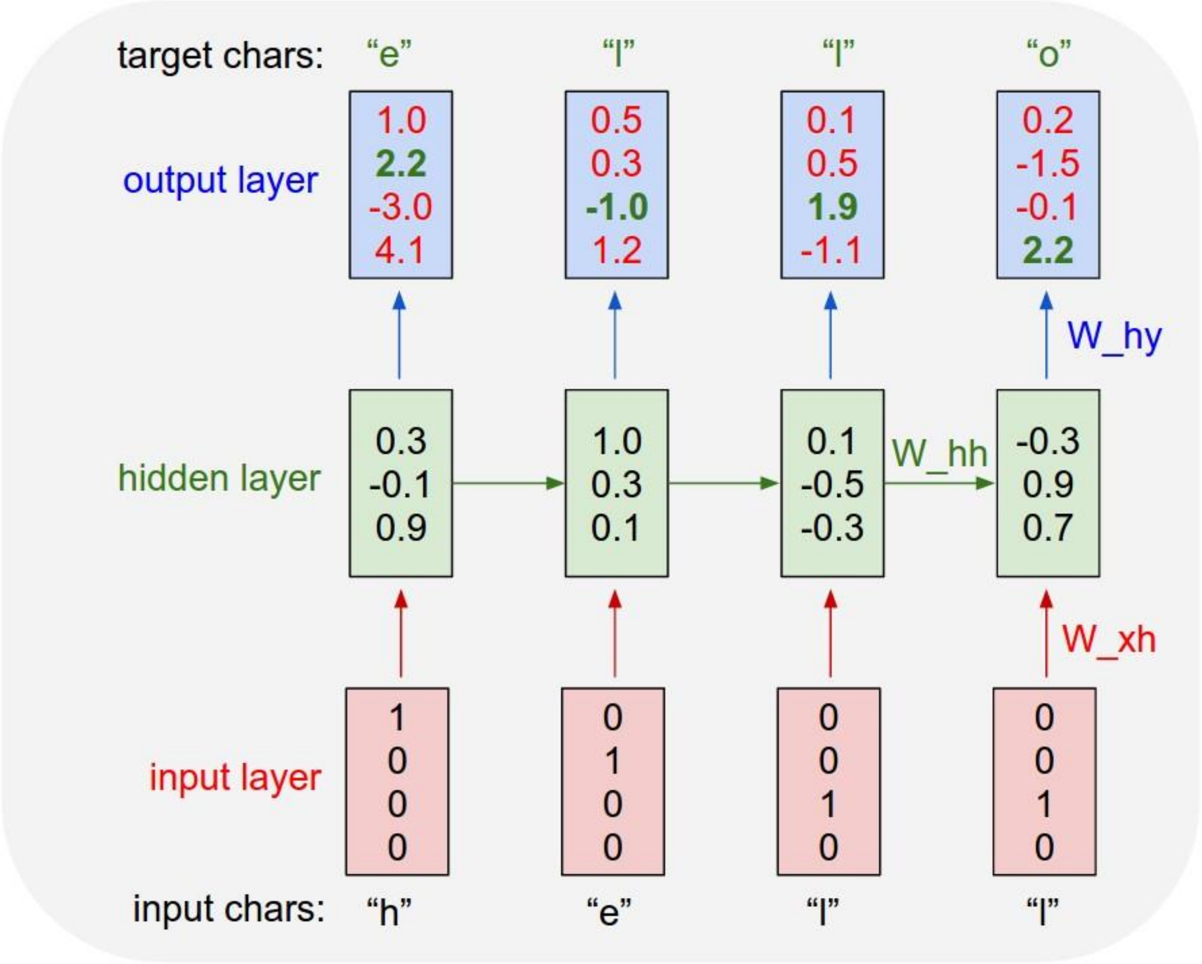
- Neural Network-based Language Model



Language Modeling

Mikolov et al. (2010)

- Recurrent Neural Network (RNN)-based Language Model
  - ✓ A simplified RNN structure for character-level language model



http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Language Modeling

- Recurrent Neural Network (RNN)-based Language Model
  - ✓ Character-level RNN vs Word-level RNN

Char-RNN

```
ESCALUS:
What is our honours, such a Richard story
Which you mark with bloody been Thilld we'll adverses:
That thou, Aurtructs a greques' great
Jmander may to save it not shif theseen my news
Clisters it take us?
Say the dulterout apy showd. They hance!

AnBESS OF GUCESTER:
Now, glarding far it prick me with this queen.
And if thou met were with revil, sir?

KATHW:
I must not my naturation disery,
And six nor's mighty wind, I fairs, if?

Messenger:
My lank, nobles arms;
```

https://github.com/hunkim/word-rnn-tensorflow

Word-RNN

```
LEONTES:
Why, my Irish time?
And argue in the lord; the man mad, must be deserved a spirit as drown the warlike Pray him, how seven in.

KING would be made that, methoughts I may married a Lord dishonour
Than thou that be mine kites and sinew for his honour
In reason prettily the sudden night upon all shalt bid him thus again. times than one from mine unaccustom'd sir.

LARTIUS:
O,'tis aediles, fight!
Farewell, it himself have saw.

SLY:
Now gods have their VINCENTIO:
Whipt fearing but first I know you you, hinder truths.

ANGELO:
This are entitle up my dearest state but deliver'd.

DUKE look dissolved: seemeth brands
That He being and
full of toad, they knew me to joy.
```

Language Modeling

Lee (2017)

- Recurrent Neural Network (RNN)-based Language Model
  - ✓ Character-level RNN (Korean)
    - 이광수 장편소설 「무정」 (총 323,660 음절, 1,680개 단어)
    - 특징: 1917년 작품이라 한자어가 많이 쓰였음, 큰따옴표와 줄바꿈을 포함한 대화체 문장이 많으며, 중고교생 대상으로 읽히는 작품이라 중간중간 괄호 속에 편집자 주석이 끼어 있음

형식은, 아뿔싸! 내가 어찌하여 이러한 생각을 하는가, 내 마음이 이렇게 약하던가 하면서 두 주먹을 불끈 쥐고 전신에 힘을 주어 이러한 약한 생각을 떼어 버리려 하나, 가슴속에는 이상하게 불길이 확확 일어난다. 이때에,
"미스터 리, 어디로 가는가" 하는 소리에 깜짝 놀라 고개를 들었다. (중략) 형식은 얼마큼 마음에 수치한 생각이 나서 고개를 돌리며, "아직 그런 말에 익숙지를 못해서……" 하고 말끝을 못 맺는다.
"대관절 어디로 가는 길인가? 급지 않거든 점심이나 하세그려."
"점심은 먹었는걸."
"그러면 맥주나 한잔 먹지."
"내가 술을 먹는가."
(중략)
"요— 오메데토오(아— 축하하네). 이이나즈케(약혼한 사람)가 있나 보네그려. 음 나루호도(그러려니). 그러구도 내게는 아무 말도 없단 말이야. 에, 여보게" 하고 손을 후려친다.

Language Modeling

Lee (2017)

Iter 0 :
랫萬게좁뉘뽈름끈玄른작밭裸觀갈나맡文플조바늠헝伍下잊볕홀툽릤혈調記운피悲렙司狼독벗칼둡걷착날完잣老
엇낫業4改'촉수륲낯깽잊쯤죽道넌友련친씌았융타雲채發造거크휘탁亨律與命텐암먼헝평琵헤落유 리벤産이馨텐

Iter 1300 : 를 옷 사가 려만다밤 말어변 대니 심로 려이, 순 과 이을 죄사글를 . 사람을 영채와 이니아베을 니러, 다가 달고 면 를 아잘 하 기 성구을 을 실튼으루 아잠 고 이 그와 매못 더 (띄어쓰기)

Iter 4900 : 를 왔다내 루방덩이종 은 얼에는 집어흔영채는 아무 우선을 에서가며 건들하아버전는 애양을 자에운 모양이 랐다. 은 한다선과 '마는 .식세식가들어 ,
형식다
"내었다.있이 문 (줄바꿈)

Iter 100000 : 면서 치현분들더 중 한통 선교잤다.
"처럼 우셨다시가...... 것이 말사도? 여자려겠습니다" 하는 마음(裸生)은 이런 적드렸**다.** 그 말이 얼굴이 딸로 나고 얼굴이 마음불 하고 따라 선

Iter 300000 : 씻었다. 선형은 형식의 형식은 빛이 가슴을 오고 걸현감에는 일이 는 눈과 의고 아이얗어 알으로 자기의 구원을 내어려가 여러 짓을 쾌처게 안아 말고였는 악한 순간에 속으로 두 학교에

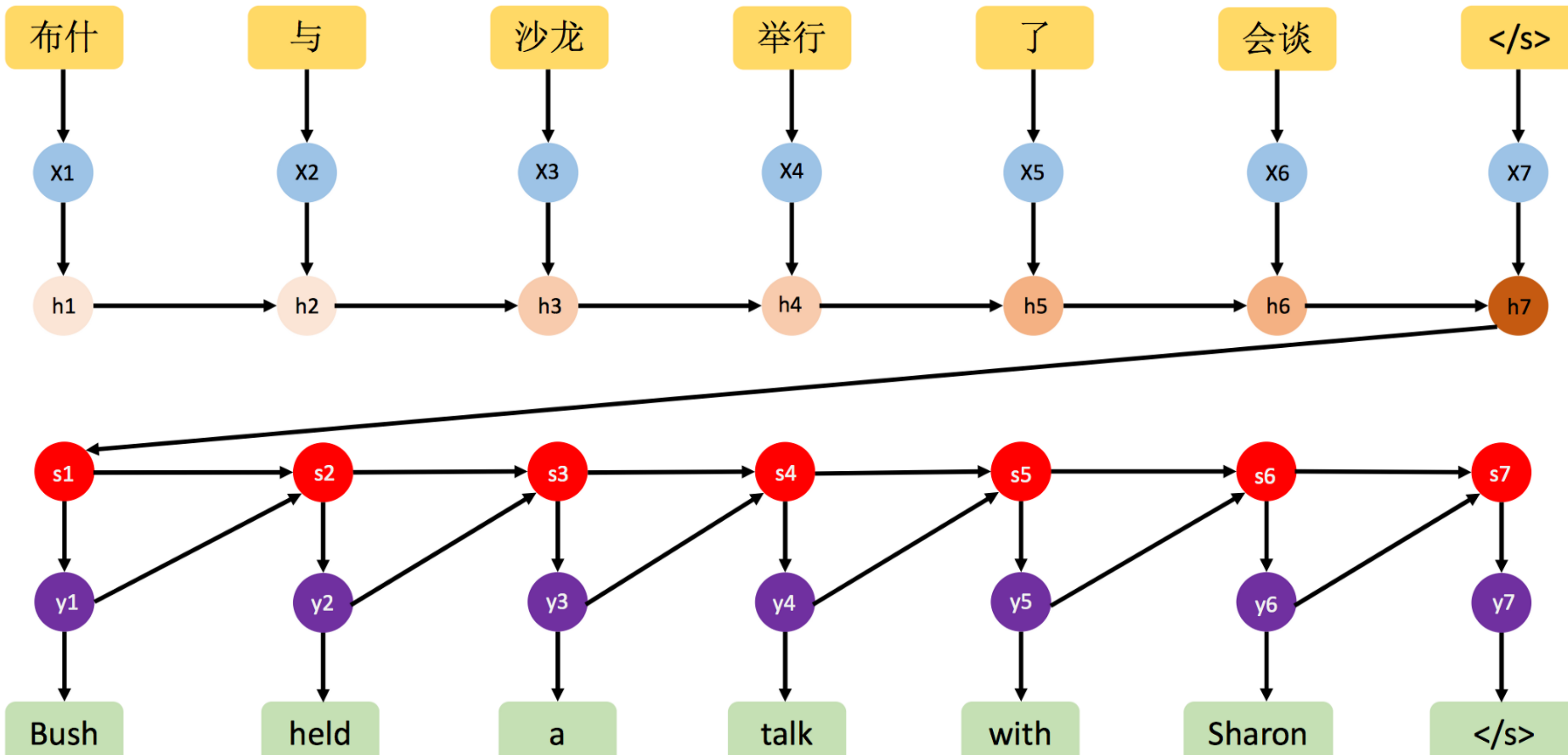
Iter 500000 : 본다. 성학과 평양으로 새로도 처음의 타던 공격하였다. '영채의 꽁인의 생각을 하면 때에 기생의 이는 것 보더니 나는 듯이 제인 소세건과 영채의 모양이를 대하였다. 형식을 생각하여

Iter 750000 : 으로 유안하였다. 더할까 하는 세상이 솔이요, 알고 게식도 들어울는 듯하였다. 태에그려 깔깔고 웃는 듯이 흔반다. 우선형은 사람을 어려보낸다.
"그려가?" (간접 인용)
한다. 영채는 손을 기쁘

Iter 1000000 : 에 돌내면서,
"여러 넣어오습데다. 그 말대 아무도 좀 집림과 시오 백매, 저는 열녀더러, 기런 소년이가 아니라."
"어리지요."
노파도 놀라며,
"저희마다가 말없습니까."
"아니 (대화체)

Language Modeling

- Sequence to Sequence (Seq2Seq) Learning



(Sutskever et al., 2014)

https://medium.com/@Synced/history-and-frontier-of-the-neural-machine-translation-dc981d25422d

Language Modeling

- Performance Improvements
  - ✓ GPT-2 (Open AI): Too Good to open the source code??

Our model, called GPT-2 (a successor to GPT), was trained simply to predict the next word in 40GB of Internet text. Due to our concerns about malicious applications of the technology, we are not releasing the trained model. As an experiment in responsible disclosure, we are instead releasing a much smaller model for researchers to experiment with, as well as a technical paper.

GPT2-Pytorch with Text-Generator

Better Language Models and Their Implications

> Our model, called GPT-2 (a successor to GPT), was trained simply to predict the next word in 40GB of Internet text. Due to our concerns about malicious applications of the technology, we are not releasing the trained model. As an experiment in responsible disclosure, we are instead releasing a much smaller model for researchers to experiment with, as well as a technical paper. from openAI Blog

This repository is simple implementation GPT-2 about **text-generator** in **Pytorch** with **compress code**

- The original repertoire is openai/gpt-2. Also You can Read Paper about gpt-2, "Language Models are Unsupervised Multitask Learners". To Understand more detail concept, I recommend papers about Transformer Model.
- Good implementation GPT-2 in Pytorch which I referred to, huggingface/pytorch-pretrained-BERT, You can see more detail implementation in huggingface repository.
- Transformer(Self-Attention) Paper : Attention Is All You Need(2017)
- First OpenAi-GPT Paper : Improving Language Understanding by Generative Pre-Training(2018)
- See OpenAI Blog about GPT-2 and Paper

https://github.com/graykode/gpt-2-Pytorch?fbclid=IwAR0mHAR1cEPpuJ7QT9TBir-37_32tkQnpMjvsu57qPacEZz2-zbTm3Iibj8

Language Modeling

- Performance Improvements
  - ✓ GPT-2 (Open AI): Too Good to open the source code??

System prompt (human-written)

`In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.`

Model completion (machine-written, 10 tries)

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."


ANY
questions?

References

Research Papers

- Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, *3*(Feb), 1137-1155.
- Bollen, J., Mao, H., Zeng, X. (2011). Twitter Mood Predicts the Stock Market, Journal of Computational Science, 2(1): 1-8.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. The Journal of Machine Learning Research, 12, 2493-2537.
- Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. Science, 349(6245), 261-266.
- Jang, M., Seo, S., & Kang, P. (2018). Recurrent Neural Network-Based Semantic Variational Autoencoder for Sequence-to-Sequence Learning, arXiv:1802.03238.
- Johnson, M., Schuster, M., Le, Q.V., Krikun, M., Wu, Y., Chen, Z., ... & Hughes, M. (2016). Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. arXiv preprint arXiv:1611.04558.
- Kim, T., Hong, J., & Kang, P. (2015). Box office forecasting using machine learning algorithms based on SNS data. *International Journal of Forecasting*, 31(2), 364-390.
- Ma, X, & Hovy, E. (2016). End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. arXiv:1603.01354.
- Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010, September). Recurrent neural network based language model. In *Interspeech* (Vol. 2, p. 3).
- Perez-Ortiz, J. A., & Forcada, M. L. (2001). Part-of-speech tagging with recurrent neural networks. Universitat d'Alacant, Spain.

References

Research Papers

- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.
- Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013, October). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)* (Vol. 1631, p. 1642).
- Strubell, E., Verga, P., Andor, D., Weiss, D., & McCallum, A. (2018). Linguistically-informed self-attention for semantic role labeling. arXiv preprint arXiv:1804.08199.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*(pp. 5998-6008).
- Wu, Yonghui, et al. "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation." *arXiv preprint arXiv:1609.08144* (2016).

References

Other materials

- Jurafsky, D. Language Modeling. Lecture Slides from the Stanford Coursera course. http://spark-public.s3.amazonaws.com/nlp/slides/languagemodeling.pdf
- Jurafsky, D. Sentiment Analysis. Lecture Slides from the Stanford Coursera course. http://spark-public.s3.amazonaws.com/nlp/slides/sentiment.pdf
- Koehn, P. (2013). Advanced Natural Language Processing Lecture 12: Parsing. School of Informatics. http://www.inf.ed.ac.uk/teaching/courses/anlp/slides/anlp12.pdf
- Koehn, P. (2013). Advanced Natural Language Processing Lecture 13: Parsing (Continued). School of Informatics. http://www.inf.ed.ac.uk/teaching/courses/anlp/slides/anlp13.pdf
- Lee, G. (2017). Recurrent neural network & long short-term memory models. DSBA Lab seminar (cs231n).
- Neubig, G. Part of Speech Tagging with Hidden Markov Models. http://www.phontron.com/slides/nlp-programming-en-05-hmm.pdf
- Witte, R. (2006). Introduction to Text Mining. Tutorial at EDBT'06. http://rene-witte.net/text-mining-tutorial-edbt2006
- http://web.stanford.edu/class/archive/cs/cs143/cs143.1128/lectures/01/Slides01.pdf