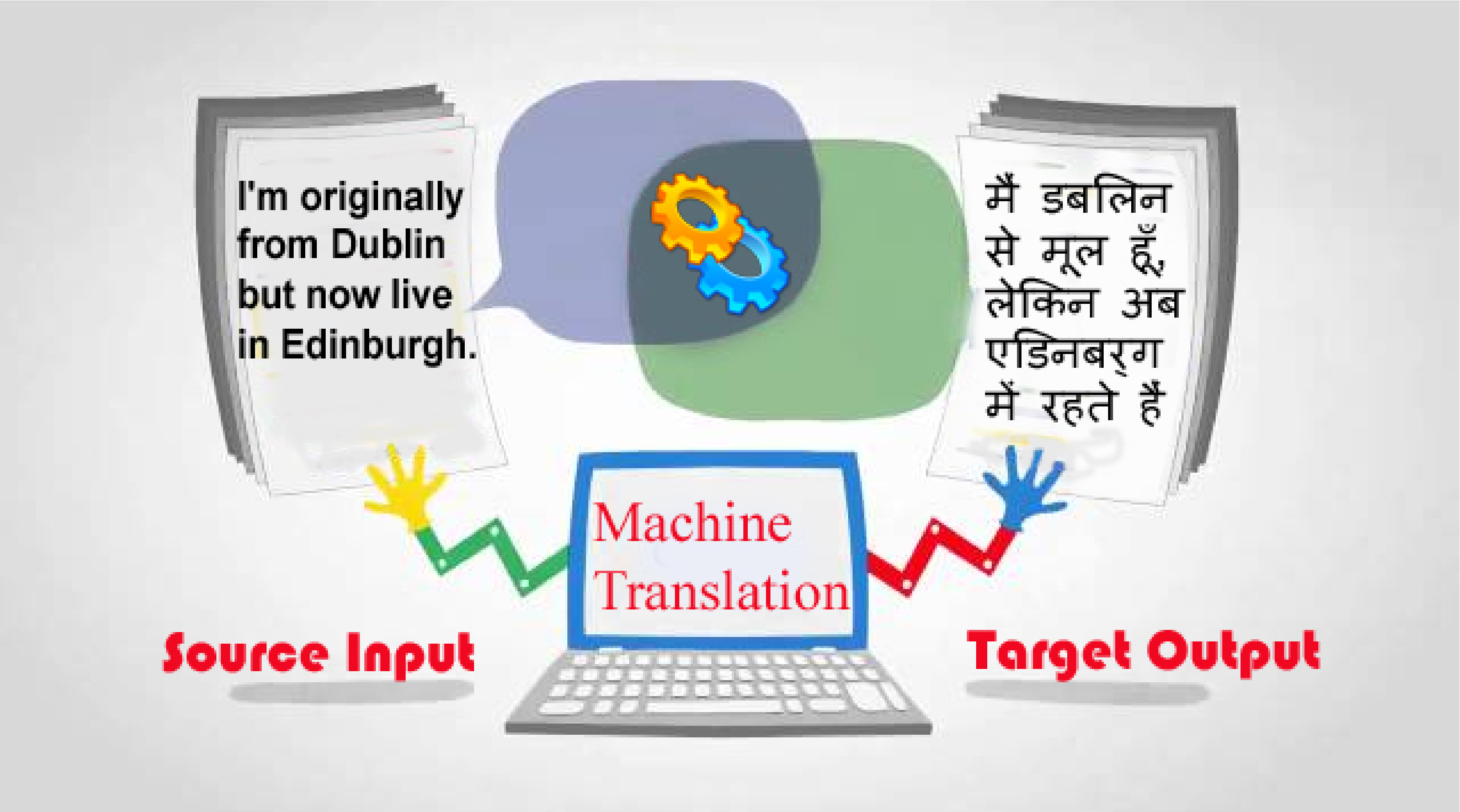
NLP와 기계번역: 통계적 기법과 머신러닝

2017년 6월 26일

강 승 식
국민대학교 소프트웨어학부

What is Machine Translation?



Topics

- MT history and NLP
- Intro. to NLP techniques
- Intro. to Machine Translation
- Statistical MT
- Machine Learning Approach

History of MT and NLP

- 7 January 1954
  - The first public demonstration of a Russian-English MT in New York, IBM
  - Having just 250 words and translating just 49 Russian sentences into English.
  - Rough translation of Russian scientific journals in order to intercept secret information.
- Early 1970s
  - Russian-English project called SYSTRAN
  - An attempt to translate a vast body of terminology connected with the military

A Critical Problem of MT

- The spirit is willing, but the flesh is weak
- The vodka is good, but the steak is lousy

The Goal of Machine Translation

강경화가 DJ 극찬받은 사연 "강이 번역하면 내 말이 더 멋있어진다"

- Automatic translation of all kinds of documents
- At a quality of the best human translators
- In fact, this goal was impossible!

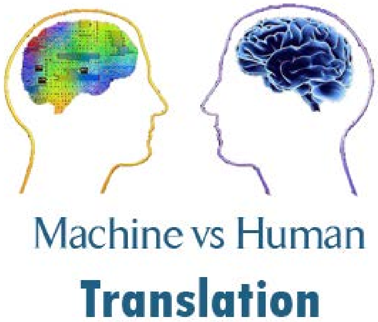
기계번역 vs. 자동통역

- 문어체 vs. 대화체
  - 문서번역 vs. 대화통역(동시 통역, 실시간 통역)

- 기계번역의 유형
  - Fully Automatic MT
  - Human-Assisted MT (HAMT)
  - Machine-Assisted Human Translation (MAHT)
    - MT Workbench

기계번역 필요성

출처: ETRI, 전자통신동향분석, 제20권 제5호, 2005년 10월.



Who is winning the race in translation?

- Google Translate

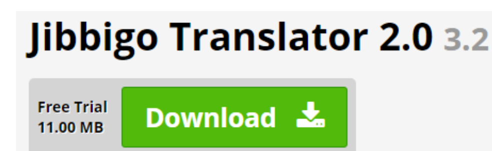
  https://translate.google.co.kr/

- Babylon

  http://translation.babylon-software.com/english/to-korean/

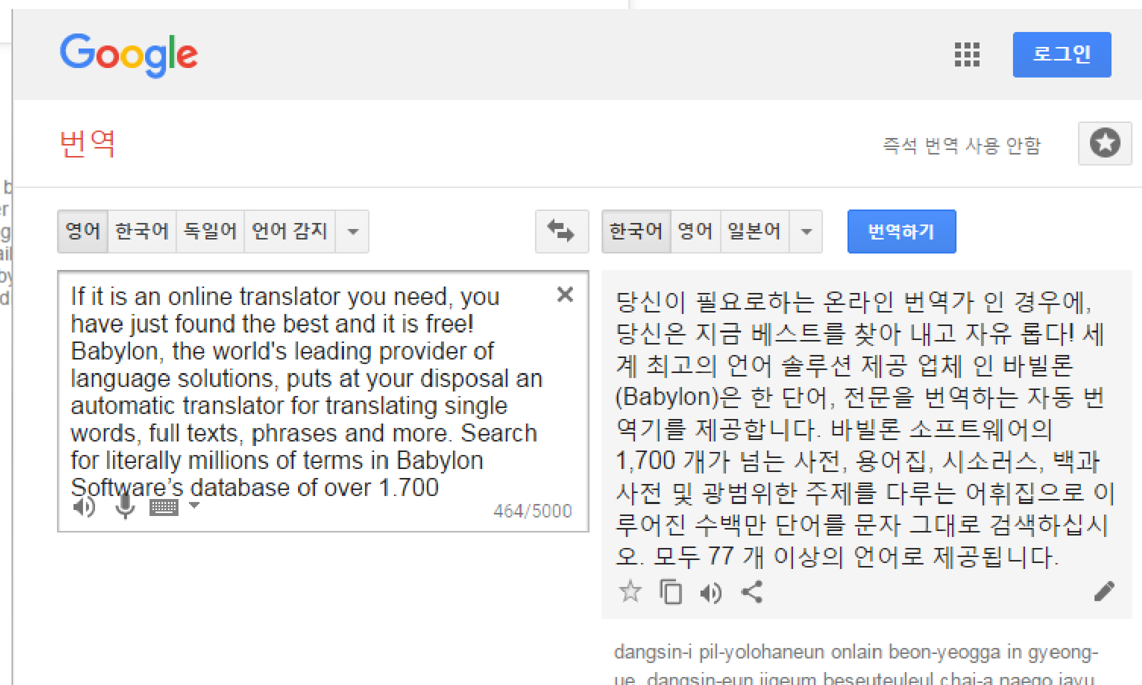
- Jibbigo

  http://jibbigo-translator-2-0.soft112.com/



- iLingual: French, German, Spanish, Arabic


Babylon 10
The most popular translation software
Download it's free
Source Language
English
Target Language
Korean
Translate
Human Translation
If it is an online translator you need, you have just found the best and it is free! Babylon, the world's leading provider of language solutions, puts at your disposal an automatic translator for translating single words, full texts, phrases and more. Search for literally millions of terms in Babylon Software's database of over 1,700 dictionaries, glossaries, thesauri, encyclopedias and lexicons covering a wide range of subjects; all in more than 77 languages.
그것은 필요한 온라인
Translator 경우 찾았
을지도 최고의 무료! 바벨론, 세계 & #039; 언어 솔루션을 공급하는 선도 업체로서 고객의 편의대로 이용하실 단일 단어를 번역하는 자동 번역, 전체 글귀, 구절 등을 배치합니다. 바빌론에서 소프트웨어의 1700여 사전, 또는 메뉴별, thesauri , 백과사전 신민들의 광범위한 용어 데이터베이스 용어 말 그대로 수백만 검색, 모두 77개 이상의 언어로.
Babylon's Free Online Translation
If it is an online English to Korean translator you need, you have just found the b
to Korean translator around, and it is free! Babylon, the world's leading provider
language solutions, puts at your disposal an automatic translator for instant Eng
Korean translation of single words and phrases. Translate documents and email
English to Korean. Search for literally millions of English to Korean terms in Baby
Software's database of over 1,700 dictionaries, glossaries, thesauri, encycloped
lexicons covering a wide range of subjects; all in more than 77 languages.


Google
로그인
번역
즉석 번역 사용 안함
영어 한국어 독일어 언어 감지
한국어 영어 일본어
번역하기
If it is an online translator you need, you have just found the best and it is free! Babylon, the world's leading provider of language solutions, puts at your disposal an automatic translator for translating single words, full texts, phrases and more. Search for literally millions of terms in Babylon Software's database of over 1.700
464/5000
당신이 필요로하는 온라인 번역가 인 경우에, 당신은 지금 베스트를 찾아 내고 자유 롭다! 세계 최고의 언어 솔루션 제공 업체 인 바빌론 (Babylon)은 한 단어, 전문을 번역하는 자동 번역기를 제공합니다. 바빌론 소프트웨어의 1,700 개가 넘는 사전, 용어집, 시소러스, 백과사전 및 광범위한 주제를 다루는 어휘집으로 이루어진 수백만 단어를 문자 그대로 검색하십시오. 모두 77 개 이상의 언어로 제공됩니다.
dangsin-i pil-yolohaneun onlain beon-yeogga in gyeong-ue, dangsin-eun jigeum beseuteuleul chaj-a naego jayu

2017-06-15

Translation Examples

- If it is an online translator you need, you have just found the best and it is free!
- 그것은 필요한 온라인 Translator 경우 찾았을지도 최고의 무료!
- 당신이 필요로하는 온라인 번역가 인 경우에, 당신은 지금 베스트를 찾아 내고 자유 롭다!
- 만약에 이것이 당신이 필요로 하는 온라인 번역기라면, 당신은 바로 가장 좋은 것을 찾았고 이것은 무료입니다.

- Babylon, the world's leading provider of language solutions,

- 바벨론, 세계 ' 언어 솔루션을 공급하는 선도 업체로서

- 세계 최고의 언어 솔루션 제공 업체 인 바빌론 (Babylon)은

- 세계의 주도적인 언어 솔루션 공급자인 바빌론은
(국제적으로 언어 솔루션을 주도적으로 공급하는 바빌론은)

- puts at your disposal an automatic translator for translating single words, full texts, phrases and more.
- 고객의 편의대로 이용하실 단일 단어를 번역하는 자동 번역, 전체 글귀, 구절 등을 배치합니다.
- 한 단어, 전문을 번역하는 자동 번역기를 제공합니다.
- 한 단어, 전문, 구 등을 번역하기 위한 자동번역기를 당신이 처분할 수 있게 해줍니다.

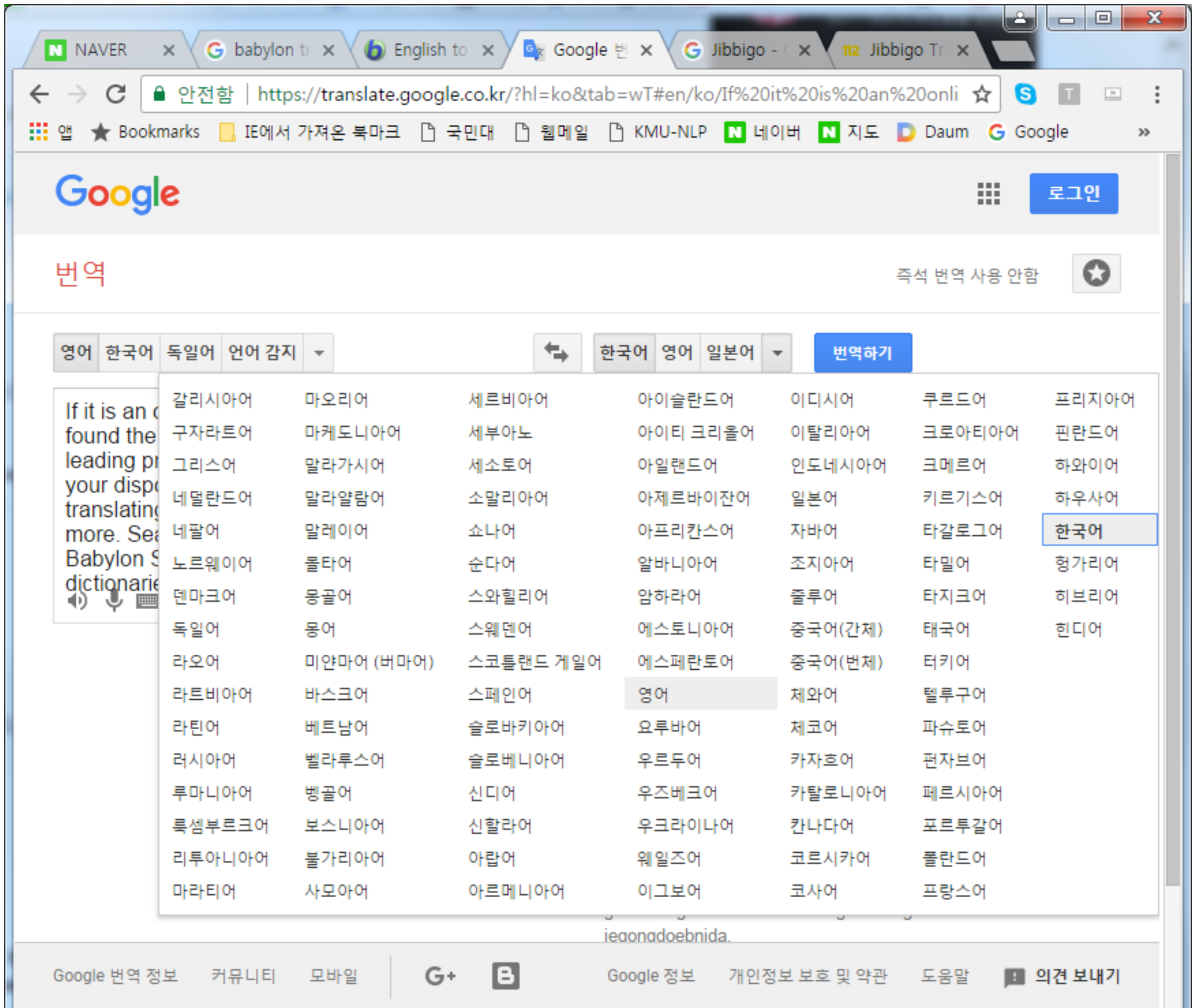
- Search for literally millions of terms in Babylon Software’s database of over 1,700 dictionaries, glossaries, thesauri, encyclopedias and lexicons covering a wide range of subjects; all in more than 77 languages.

- 바빌론에서 소프트웨어의 1700여 사전, 또는 메뉴별, thesauri , 백과사전 신민들의 광범위한 용어 데이터베이스 용어 말 그대로 수백만 검색, 모두 77개 이상의 언어로.

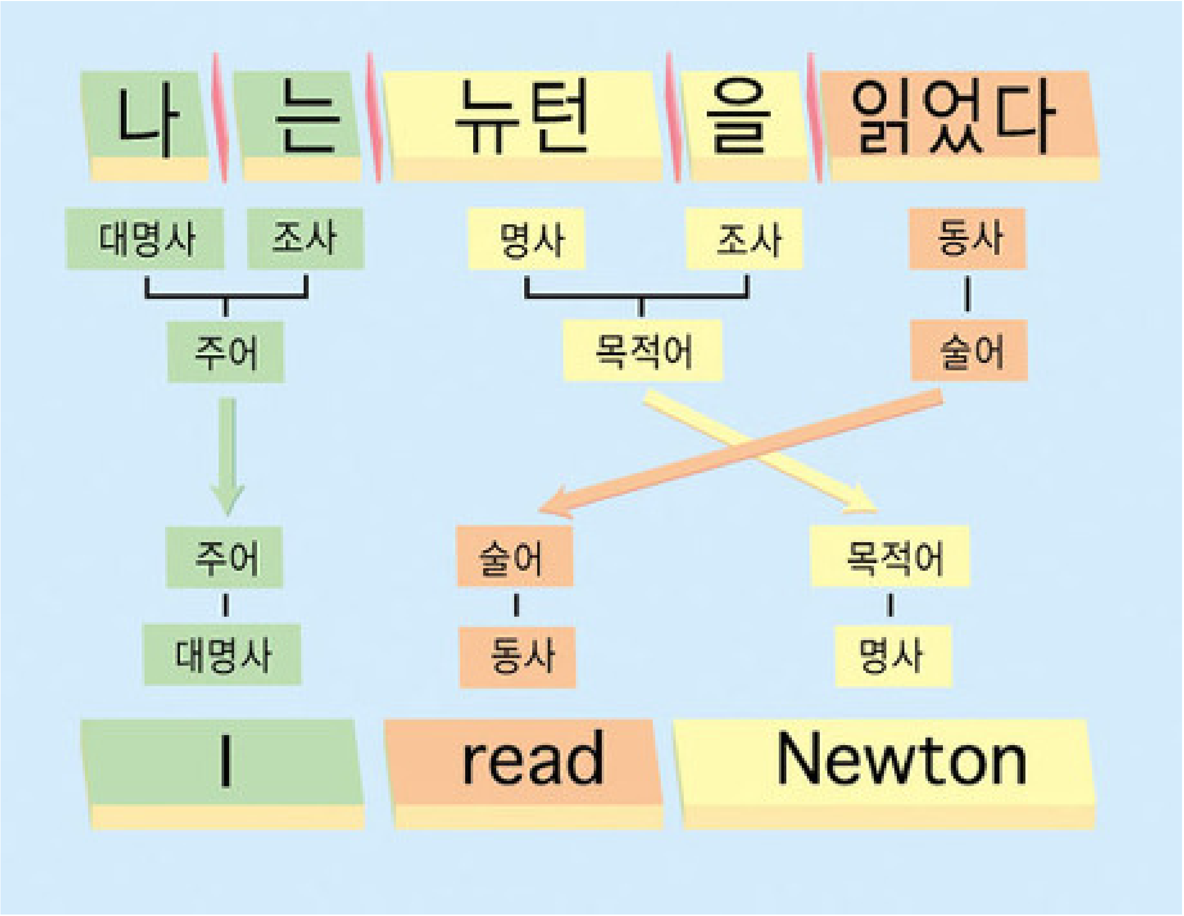
- 바빌론 소프트웨어의 1,700 개가 넘는 사전, 용어집, 시소러스, 백과 사전 및 광범위한 주제를 다루는 어휘집으로 이루어진 수백만 단어를 문자 그대로 검색하십시오. 모두 77 개 이상의 언어로 제공됩니다.

- 넓은 범위의 주제들을 포괄하는 1,700개 이상의 사전과 용어집, 시소러스, 백과사전, 어휘사전을 보유하고 있는 바빌론 소프트웨어 데이터베이스에서 수백만개의 용어들을 검색해 보세요. 모두 77개 이상의 언어로.

How many languages? 104

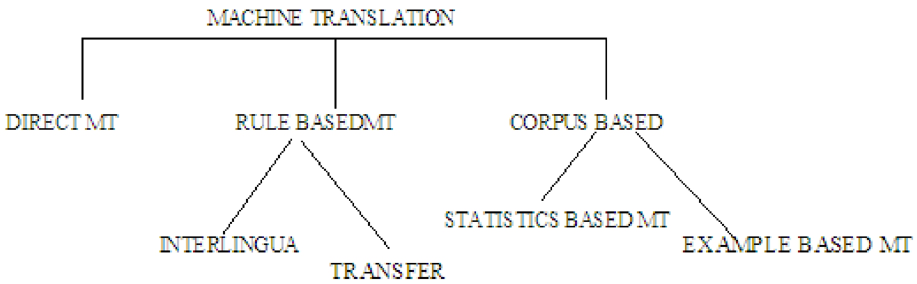


기계번역 방법 예제



나
는
뉴턴
을
읽었다
대명사
조사
명사
조사
동사
주어
목적어
술어
주어
술어
목적어
대명사
동사
명사
I
read
Newton

M.T. Approaches



- Direct Translation
- Rule-Based M.T.
  - Transfer-based Approach
  - Interlingua/Pivot Approach
- Corpus-Based M.T.
  - Statistical M.T. (SMT)
  - Example-Based M.T. (EBMT)
- Knowledge-Based M.T.
- Neural Network Approach

Traditional MT approaches

- Transfer-based
- Interlingua
- Example-based (EBMT)
- Statistical MT (SMT)
- Hybrid approach

Direct Translation

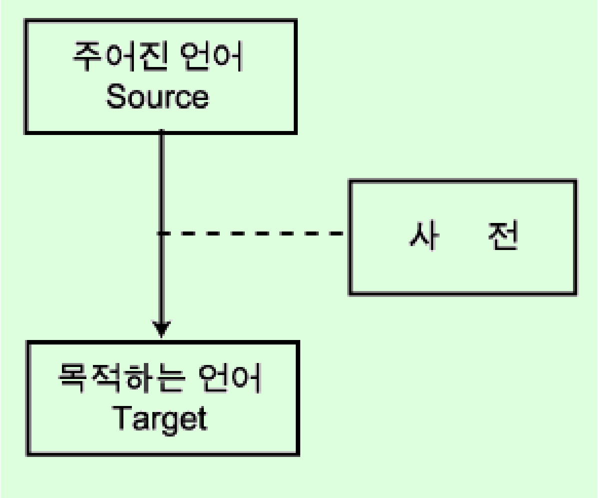


그림 1 직접변환 방식

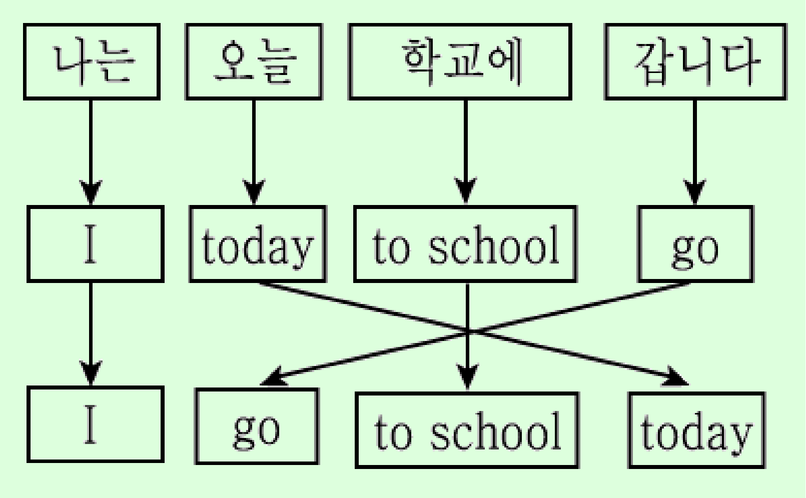


그림 2 직접변환방식에 의한 번역

- 인공지능 입문 - 그림으로 풀어본 : 도우치 준이치 지음, 최기선 옮김, 미래사, 1992, Page 129~141

Transfer Approach

- Number of translators: N x N

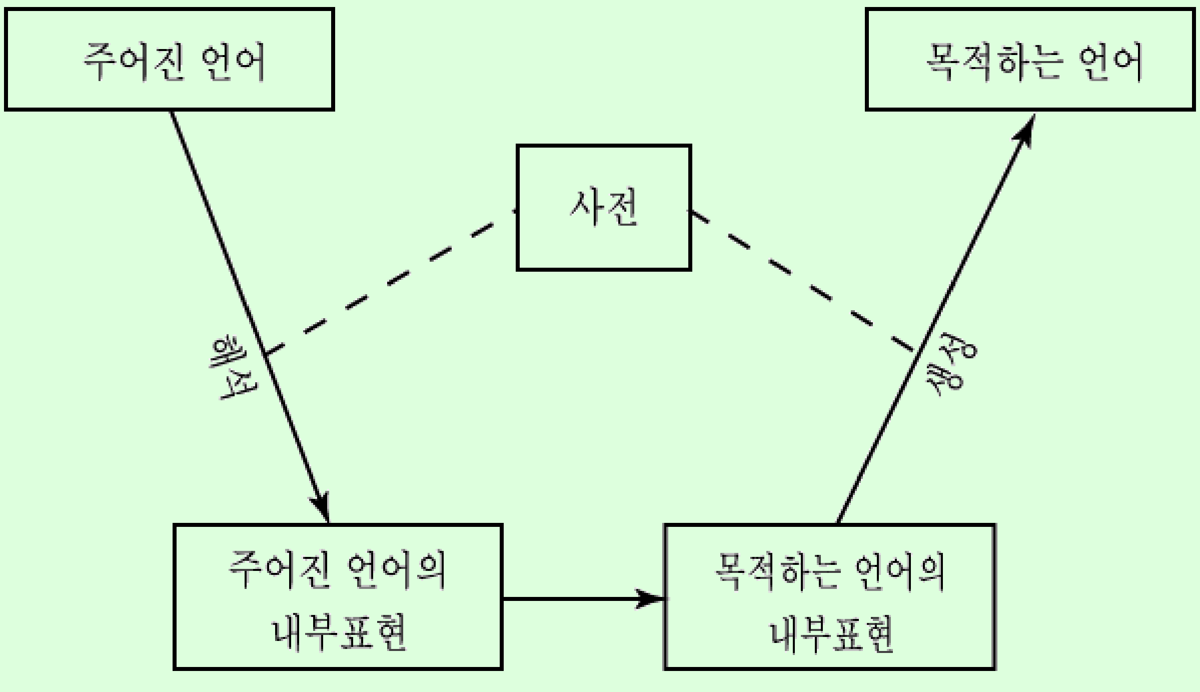


그림 5 간접변환 방식

- Analysis, transfer, generation:
    1. Parse the source sentence
    2. Transform the parse tree with transfer rules
    3. Translate source words
    4. Get the target sentence from the tree

- Resources required:
    - Source parser
    - A translation lexicon
    - A set of transfer rules

Example: Korean-to-English

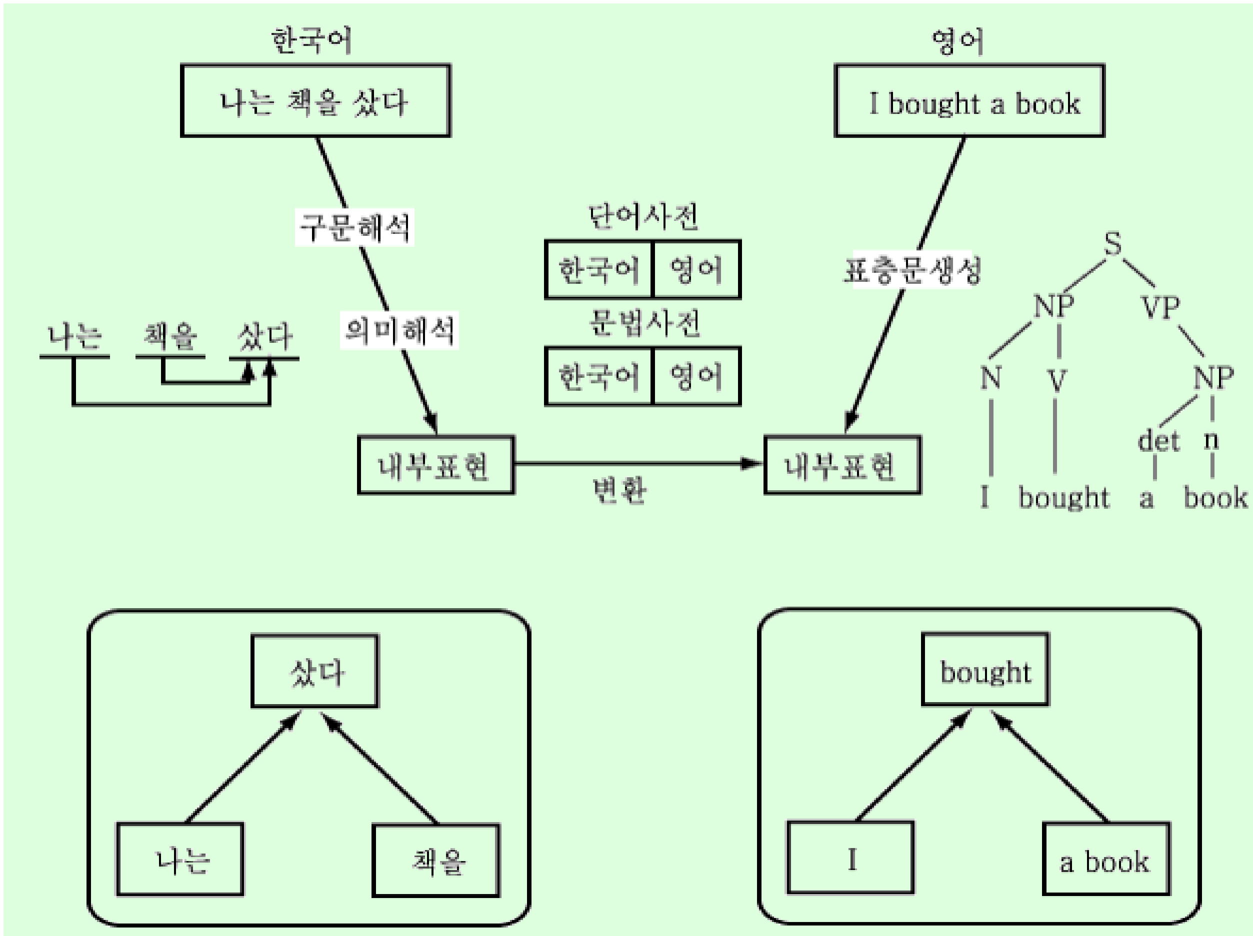


그림 7 간접변환방식에 의한 번역

Issues in Transfer-based MT

- Parsing: linguistically motivated grammar or formal grammar?
- Transfer:
  - context-free rules? A path on a dependency tree?
  - Apply at most one rule at each level?
  - How are rules created?
- Translating words: word-to-word translation?
- Generation: using LM or other additional knowledge?

- How to create the needed resources automatically?
- For n languages, we need n(n-1) MT systems!

Interlingua Approach

- Language-independent representation of a sentence

- We only need *n* analyzers, and *n* generators.

- Resource needed:
  - A language-independent representation
  - Sophisticated analyzers
  - Sophisticated generators

Interlingua/Pivot Approach

- Esperanto like intermediate representation

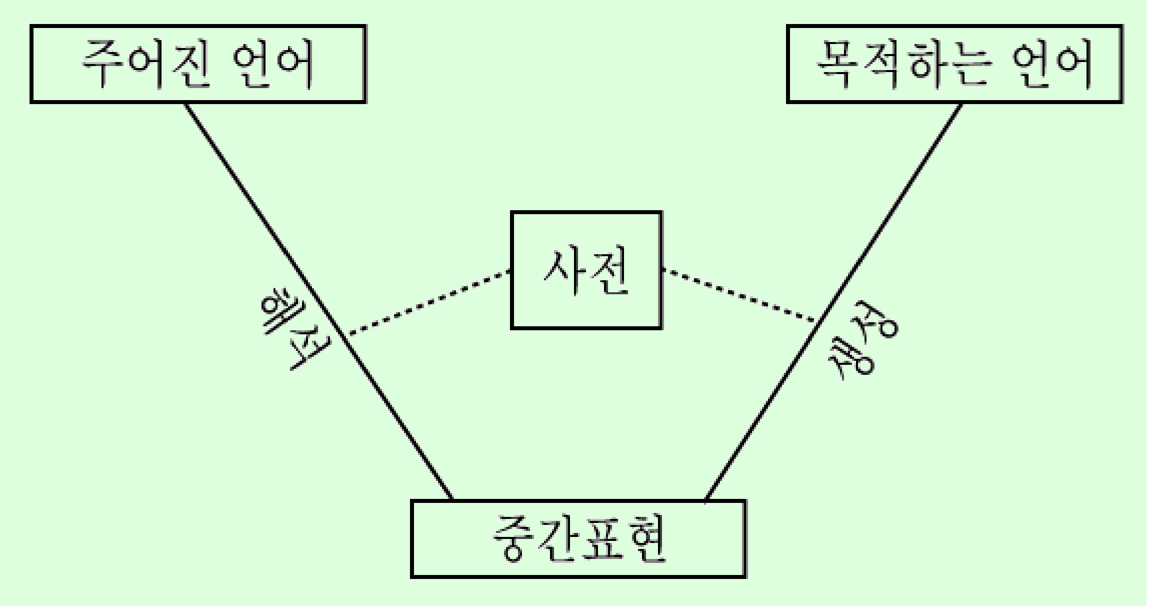


그림 3 피벗 방식

Analysis & Generation

- Number of translators: N + N

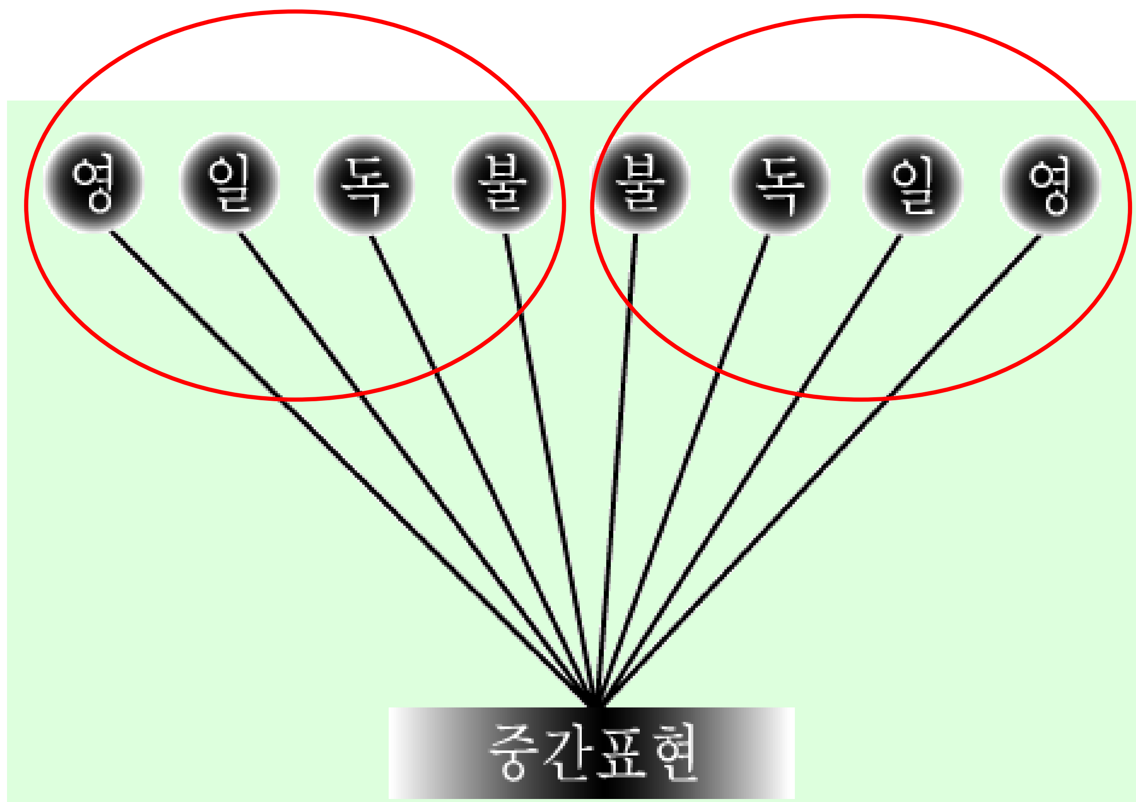
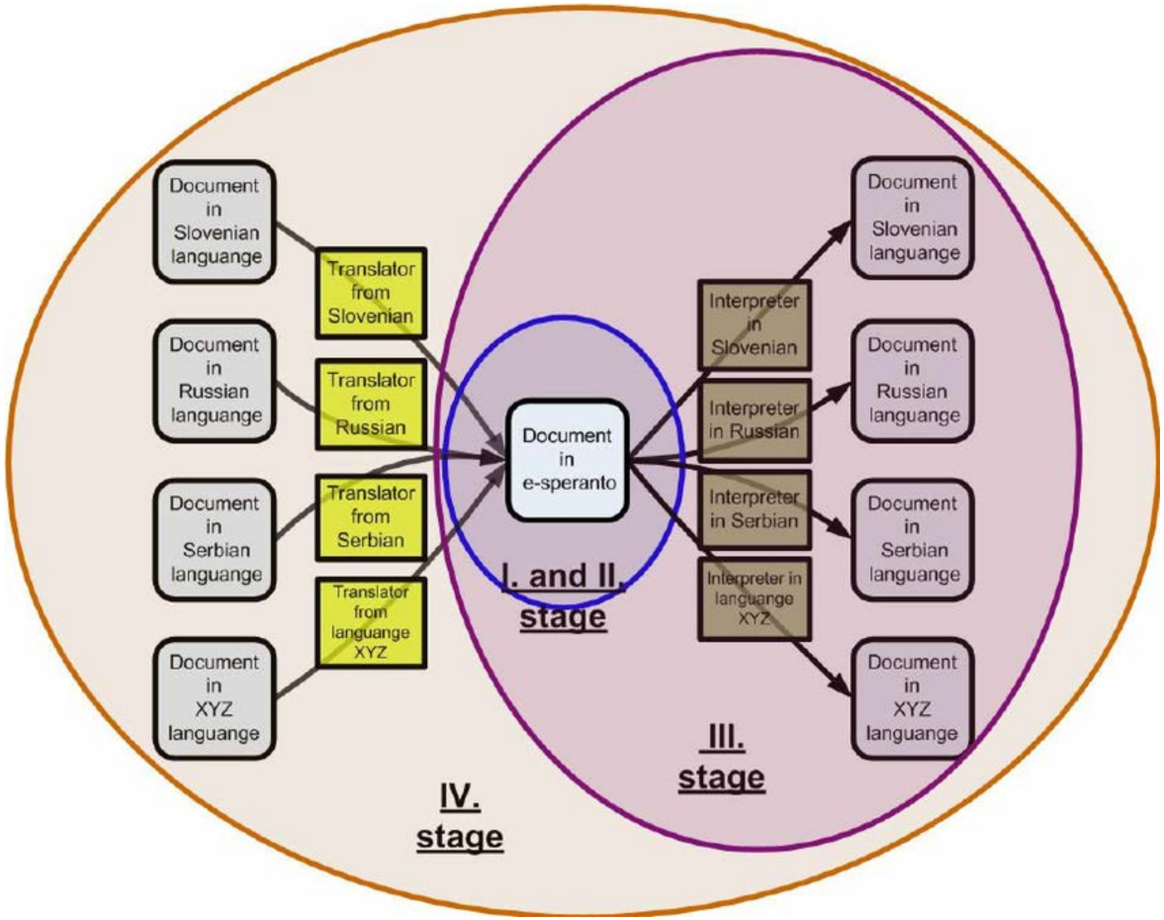


그림 4 다언어간 번역

Interlingua: Pivot Approach



Direct, Transfer, and Interlingua

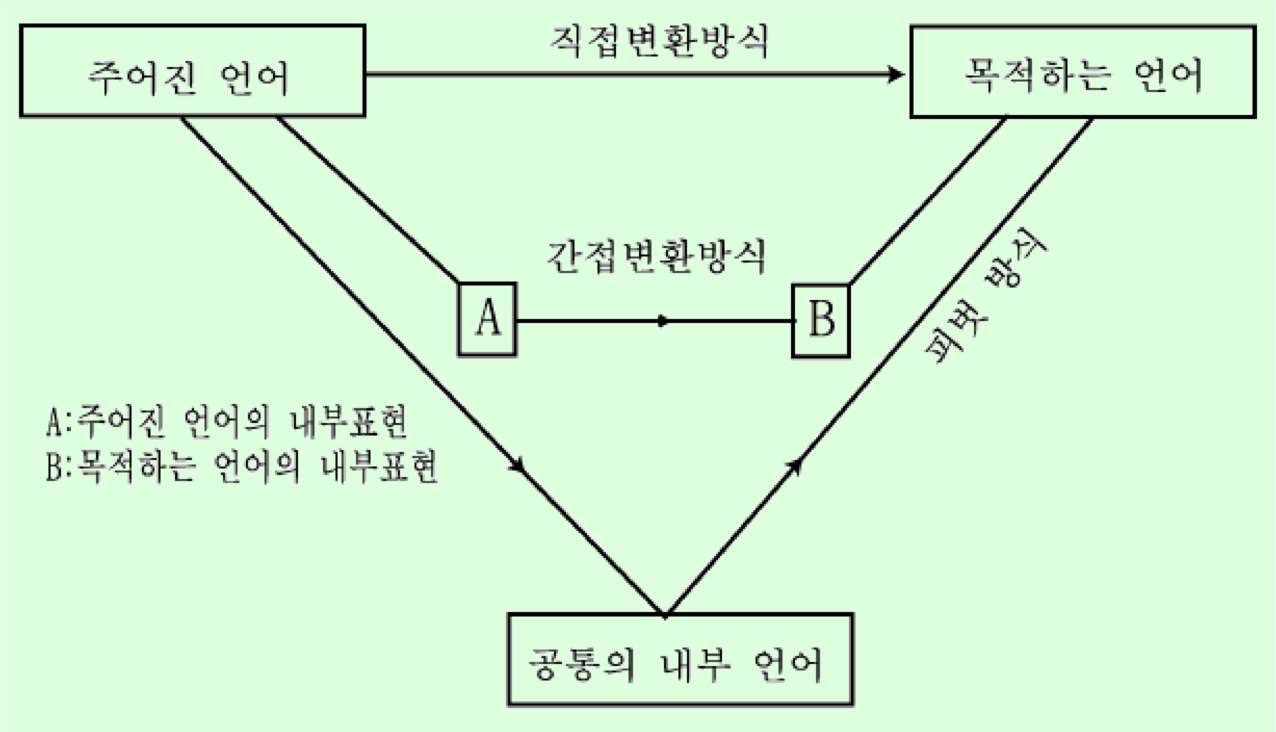
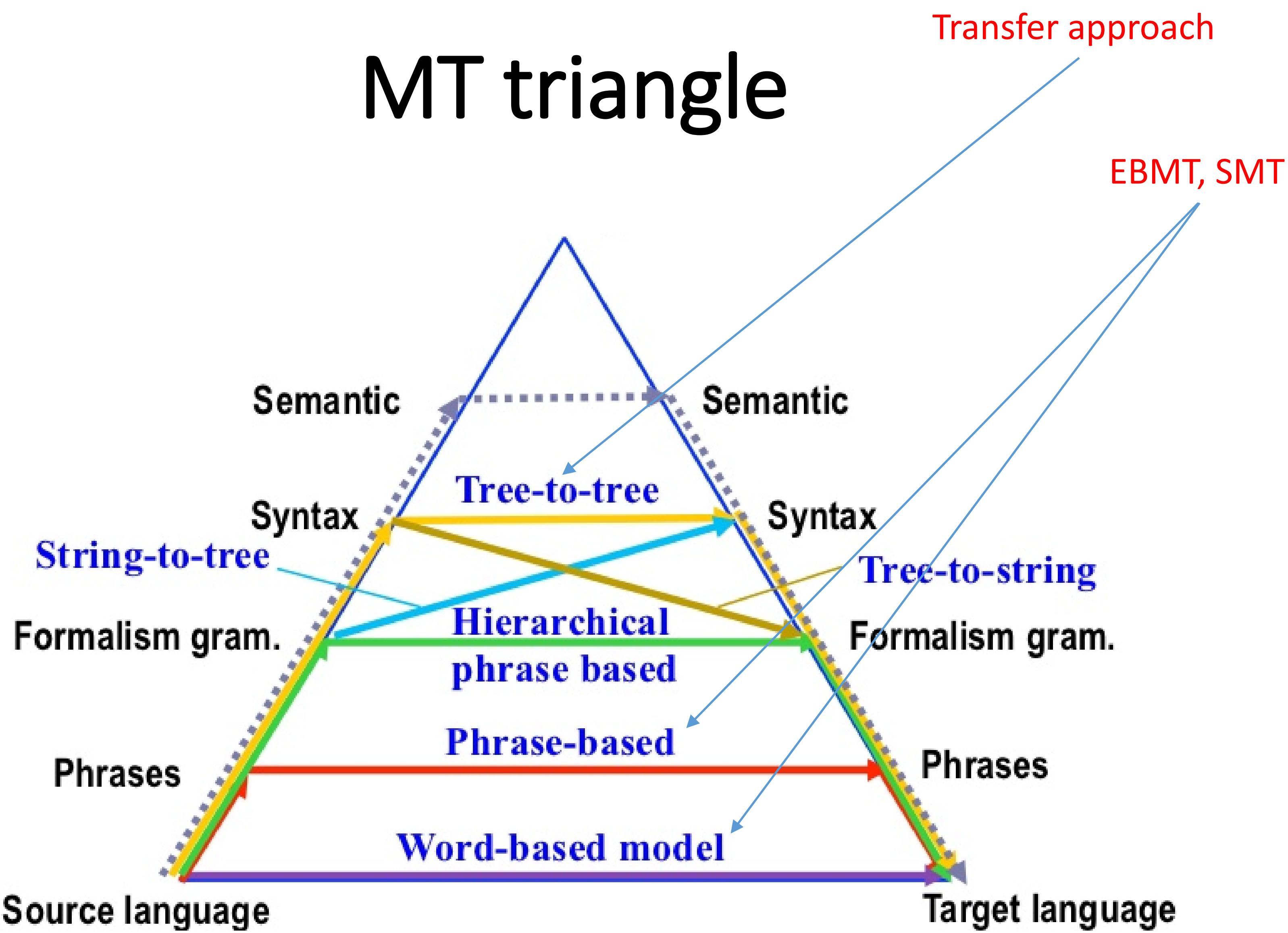


그림 6 3개의 변환방식

MT triangle


Transfer approach
EBMT, SMT
Semantic
Semantic
Tree-to-tree
Syntax
Syntax
String-to-tree
Tree-to-string
Formalism gram.
Hierarchical
phrase based
Formalism gram.
Phrase-based
Phrases
Phrases
Word-based model
Source language
Target language

Issues in Interlingua

- Language-independent meaning representation really exist? If so, what does it look like?
- It requires deep analysis: how to get such an analyzer: e.g., semantic analysis
- It requires non-trivial generation: How is that done?
- It forces disambiguation at various levels: lexical, syntactic, semantic, discourse levels.

NLP and Machine Translation

is to

Analysis and Generation

NLP issues and applications

NLP Basics

- Morphological analysis(형태소 분석)
  - Word-level
- Syntactic analysis(구문 분석)
  - Sentence-level
- Semantic analysis(의미 분석)
  - Word-sense disambiguation
- Natural Language Generation(자연어 생성)
- Language Resources(언어 자원)
  - 말뭉치, WordNet, 온톨로지 등

NLP Applications

- Machine Translation, 1950's-now
- Information Retrieval, 1980's-now
  - Text Classification, Information Extraction
  - Text Summarization
  - Text Mining, Opinion Mining
  - Sentiment Classification(감성 분류)
- Natural Language Understanding, 1960-70, 2000's
  - ELIZA: Doctor, Joseph Weizenbaum, MIT, 1965
  - SHRDLU: Robot arm, Terry Winograd, MIT, 1971
  - LUNAR
  - Ask Jeeves(ask.com), 1996
  - Wolfram alpha, 2009

- Speller and grammar checker

- Spam mail filtering, Spam 문자 filtering
- Sentiment analysis(감성 분석)

- 아이폰 시리, IBM 왓슨, 자동통역 시스템
- 텍스트 마이닝, 빅데이터 분석

NLP Resources and NLTK in Python

NLP resources in http://nlp.stanford.edu/

Core | Projects | Archive

- + Stanford CoreNLP
- + Stanford Parser
- + Stanford POS Tagger
- + Stanford Named Entity Recognizer
- + Stanford RegexNER
- + Stanford Word Segmenter
- + Stanford Classifier
- + Stanford EnglishTokenizer
- + Stanford TokensRegex
- + Stanford Temporal Tagger (SUTime)
- + Stanford Pattern-based Information Extraction and Diagnostics (SPIED)
- + Stanford Relation Extractor

Core | Projects | Archive

+ Stanford Neural Machine Translation

+ Stanford Natural Language Inference Corpus (SNLI)

+ Semantic Parsing with Execution (SEMPRE)

SEMPRE is a toolkit for training semantic parsers, which map natural language utterances to denotations (answers) via intermediate logical forms.

+ Stanford Open Information Extraction

A tool for extracting open domain relation triples; e.g., "cats play with yarn" yields (cats; play with; yarn).

+ GloVe: Global Vectors for Word Representations

+ Deep Learning for Sentiment Analysis

+ Tregex, Tsurgeon, and Semgrex

Tools for matching patterns in linguistic trees (following the tgrep/tgrep2 tradition), a GUI for this, and a tree-transformation utility built on top of this matching language. Also, a similar utility for matching patterns in dependency graphs.

+ Phrasal

A state-of-the-art phrase-based machine translation system.

POS tagging

The strongest rain ever recorded in India shut down the financial hub of Mumbai, snapped communication lines, closed airports and forced thousands of people to sleep in their offices or walk home during the night, officials said today.

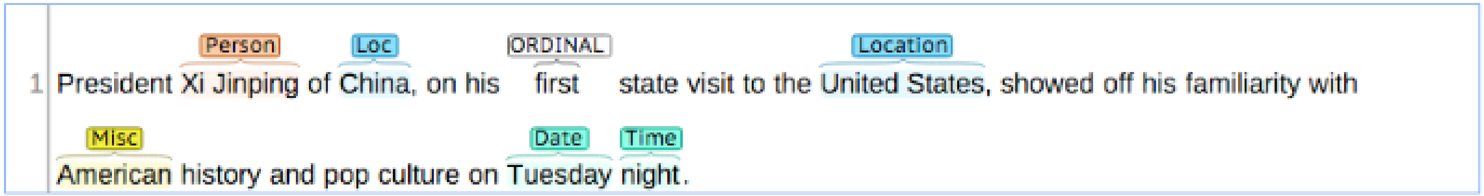
The/DT strongest/JJS rain/NN ever/RB recorded/VBN in/IN India/NNP shut/VBD down/RP the/DT financial/JJ hub/NN of/IN Mumbai/NNP ,/, snapped/VBD communication/NN lines/NNS ,/, closed/VBD airports/NNS and/CC forced/VBD thousands/NNS of/IN people/NNS to/TO sleep/VB in/IN their/PRP$ offices/NNS or/CC walk/VB home/NN during/IN the/DT night/NN ,/, officials/NNS said/VBD today/NN ./.

```
(ROOT
  (S
    (S
      (NP
        (NP (DT The) (JJS strongest) (NN rain))
        (VP
          (ADVP (RB ever))
          (VBN recorded)
          (PP (IN in)
            (NP (NNP India)))))
      (VP
        (VP (VBD shut)
          (PRT (RP down))
          (NP
            (NP (DT the) (JJ financial) (NN hub))
            (PP (IN of)
              (NP (NNP Mumbai)))))
        (, ,)
        (VP (VBD snapped)
          (NP (NN communication) (NNS lines)))
        (, ,)
        (VP (VBD closed)
          (NP (NNS airports)))
        (CC and)
        (VP (VBD forced)
          (NP
            (NP (NNS thousands))
            (PP (IN of)
              (NP (NNS people))))
          (S
            (VP (TO to)
              (VP
                (VP (VB sleep)
                  (PP (IN in)
                    (NP (PRP$ their) (NNS offices))))
                (CC or)
                (VP (VB walk)
                  (NP (NN home))
                  (PP (IN during)
                    (NP (DT the) (NN night))))))))))
    (, ,)
    (NP (NNS officials))
    (VP (VBD said)
      (NP-TMP (NN today)))
    (. .)))
```

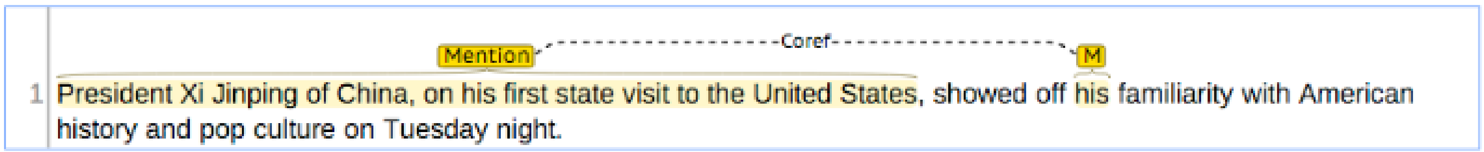
```
det(rain-3, The-1)
amod(rain-3, strongest-2)
nsubj(shut-8, rain-3)
nsubj(snapped-16, rain-3)
nsubj(closed-20, rain-3)
nsubj(forced-23, rain-3)
advmod(recorded-5, ever-4)
partmod(rain-3, recorded-5)
prep_in(recorded-5, India-7)
ccomp(said-40, shut-8)
prt(shut-8, down-9)
det(hub-12, the-10)
amod(hub-12, financial-11)
dobj(shut-8, hub-12)
prep_of(hub-12, Mumbai-14)
conj_and(shut-8, snapped-16)
ccomp(said-40, snapped-16)
nn(lines-18, communication-17)
dobj(snapped-16, lines-18)
conj_and(shut-8, closed-20)
ccomp(said-40, closed-20)
dobj(closed-20, airports-21)
conj_and(shut-8, forced-23)
ccomp(said-40, forced-23)
dobj(forced-23, thousands-24)
prep_of(thousands-24, people-26)
aux(sleep-28, to-27)
xcomp(forced-23, sleep-28)
poss(offices-31, their-30)
prep_in(sleep-28, offices-31)
xcomp(forced-23, walk-33)
conj_or(sleep-28, walk-33)
dobj(walk-33, home-34)
det(night-37, the-36)
prep_during(walk-33, night-37)
nsubj(said-40, officials-39)
root(ROOT-0, said-40)
tmod(said-40, today-41)
```

- This output was generated with the command:
- java -mx200m edu.stanford.nlp.parser.lexparser.LexicalizedParser -retainTMPSubcategories -outputFormat "wordsAndTags,penn,typedDependencies" englishPCFG.ser.gz mumbai.txt

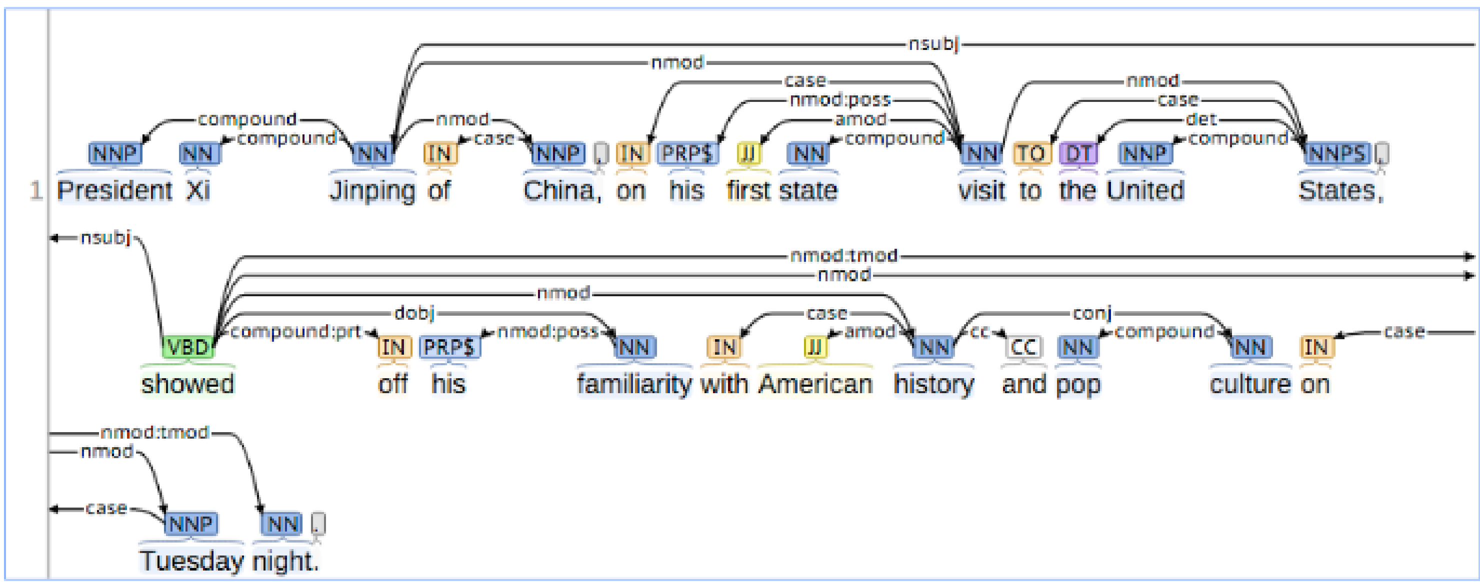
Named Entity Recognition:



Coreference:



Basic Dependencies:



NLTK: NLP Took Kit

- Natural Language Toolkit
  - http://www.nltk.org/
- Suite of classes for several NLP tasks
  - Parsing, POS tagging, classifiers…
- Easy-to-use interfaces to over 50 corpora and lexical resources
  - http://www.nltk.org/nltk_data/

Installing NLTK

- http://www.nltk.org/install.html

- Mac/Unix
  1. Install Setuptools
  2. Install Pip
  3. Install Numpy(optional)
  4. Install PyYAML and NLTK
  5. Test installation

- Windows
  1. Install Python
  2. Install Numpy(optional)
  3. Install Setuptools
  4. Install Pip
  5. Install PyYAML and NLTK
  6. Test installation

Modules

- The NLTK modules include:
  - nltk.token : processing individual elements of text, such as words or sentences
  - nltk.tagger : tagging tokens with supplemental information, such as POS or wordnet sense tags
  - nltk.parser : high-level interface for parsing texts
  - nltk.classify : classify text into categories
  - nltk.corpus : access (tagged)corpus data

....

- http://www.nltk.org/py-modindex.html#

Example: POS tagging

POS_tagging.py - C:₩Users₩jinwoo₩Desktop₩python_ex₩POS_tagging.py

File Edit Format Run Options Windows Help

```python
from nltk import pos_tag,word_tokenize
sentent1 = '''
this is a demo that will show you how to
detects parts of speech with little effort using NLTK!'
'''
tokenized_sent = word_tokenize(sentent1)
print pos_tag(tokenized_sent)
```

Ln: 4 Col: 0

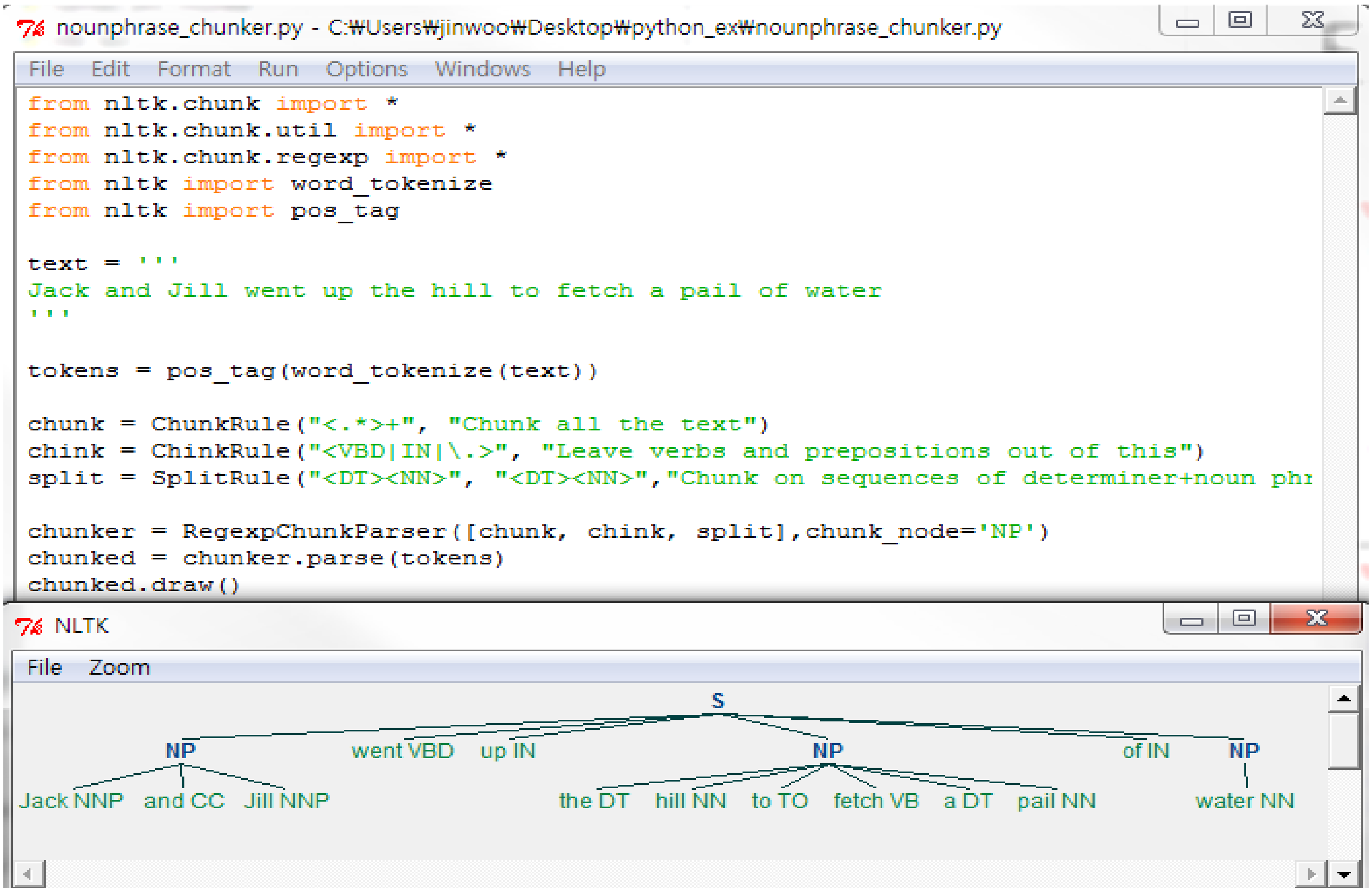
Python Shell

File Edit Shell Debug Options Windows Help

```
alicia ambiguity anger bathos blunder boards commoners correspondence
doors english france incubi it james organ oriental ours outspoken
paper perpetration
>>> ================================ RESTART ================================
>>>
[('this', 'DT'), ('is', 'VBZ'), ('a', 'DT'), ('demo', 'NN'), ('that', 'WDT'), ('
will', 'MD'), ('show', 'VB'), ('you', 'PRP'), ('how', 'WRB'), ('to', 'TO'), ('de
tects', 'NNS'), ('parts', 'NNS'), ('of', 'IN'), ('speech', 'NN'), ('with', 'IN')
, ('little', 'JJ'), ('effort', 'NN'), ('using', 'VBG'), ('NLTK', 'NNP'), ('!', '
.'), ("'", "'''")]
>>>
```

Ln: 214 Col: 44

Example: Parsing



Example: WordNet

similarity.py - C:₩Users₩jinwoo₩Desktop₩python_ex₩similarity.py

```python
from nltk.corpus import wordnet as wn

Aword = 'language'
Bword = 'barrier'

synsetsA = wn.synsets(Aword)
synsetsB = wn.synsets(Bword)

similars = []

for sseta in synsetsA:
        for ssetb in synsetsB:
                path_similarity = sseta.path_similarity(ssetb)

                if path_similarity is not None:
                        similars.append({
                                'path':path_similarity,
                                'wordA':sseta,
                                'wordB':ssetb,
                                'wordA_definition':sseta.definition,
                                'wordB_definition':ssetb.definition
                        })

similars = sorted(similars, key=lambda item: item['path'],reverse=True)

for item in similars:
        print item['wordA'],"\n",item['wordA_definition']
        print item['wordB'],"\n",item['wordB_definition']
        print 'Path similarity - ',item['path'],"\n"
```

Ln: 30 Col: 0

Python Shell

```
>>> ================================ RESTART ================================
>>>
Synset('linguistic_process.n.02')
the cognitive processes involved in producing and understanding linguistic commu
nication
Synset('barrier.n.02')
any condition that makes it difficult to make progress or to achieve an objectiv
e
Path similarity -  0.111111111111

Synset('language.n.05')
the mental faculty or power of vocal communication
Synset('barrier.n.02')
any condition that makes it difficult to make progress or to achieve an objectiv
e
Path similarity -  0.111111111111

Synset('language.n.01')
a systematic means of communicating by the use of sounds or conventional symbols
Synset('barrier.n.02')
any condition that makes it difficult to make progress or to achieve an objectiv
e
Path similarity -  0.1

Synset('language.n.01')
a systematic means of communicating by the use of sounds or conventional symbols
Synset('barrier.n.03')
anything serving to maintain separation by obstructing vision or access
Path similarity -  0.1

Synset('language.n.01')
a systematic means of communicating by the use of sounds or conventional symbols
Synset('barrier.n.01')
a structure or object that impedes free movement
Path similarity -  0.0909090909091
```

Ln: 431 Col: 35

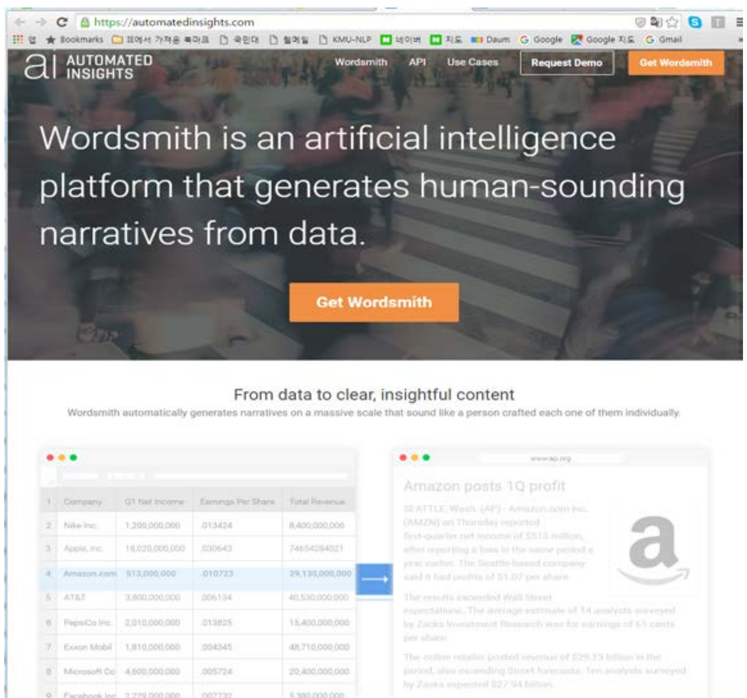
For more details

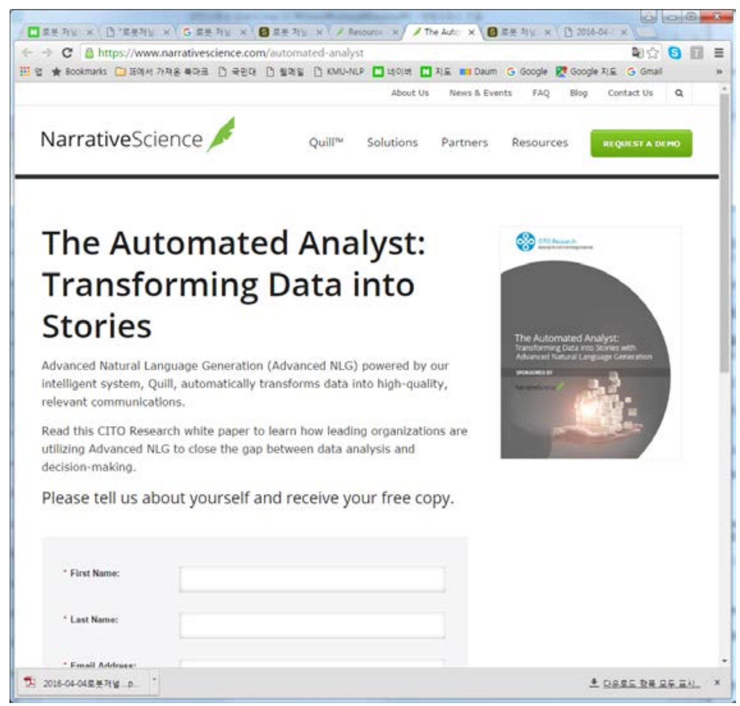
- NLTK
  - http://www.nltk.org/index.html

- NLTK demo site
  - http://text-processing.com/demo/

NLP Generation

- Robot Journalism: 스포츠, 지진, 교통, 일기예보
  - https://automatedinsights.com/
  - https://www.narrativescience.com/





NLP Generation (cont)

- ChatBot: dialogue analysis and generation

- Pattern match in the new programming languages
  - Scala, Swift, and Wolfram Language

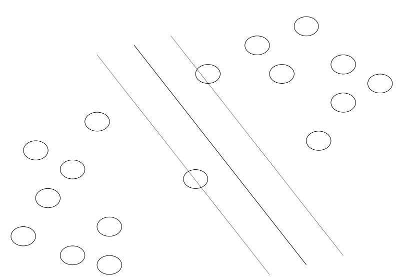
NLP, Machine Learning, and Machine Translation

Machine Learning for NLP

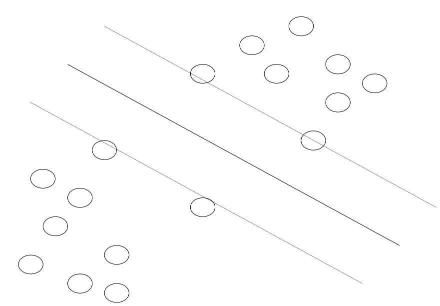
- HMM, MEM(Maximum Entropy Model)
- kNN(k-Nearest Neighbor)
- Naïve Bayse
- SVM(Support Vector Machine)
- CRF++ (Conditional Random Field)
- Neural Network

Support Vector Machine (SVM)

- Support Vector Machine (SVM)
  - 이원(binary) 패턴 인식 문제를 해결하기 위해 제안된 학습 방법
  - 두 클래스 사이에 가장 최적의 결정면(벡터 평면)을 찾는 것이 목적



smaller margin



maximal margin

SVM: binary classifier

- SVM light
  - Thorsten Joachims <thorsten@joachims.org>
    - Cornell University Department of Computer Science
  - An implementation of the SVMs in C.

- SVM 엔진 다운로드
  - http://svmlight.joachims.org/
  - source code: http://download.joachims.org/svm_light/current/svm_light.tar.gz
  - Binary versions are also available for the various systems.

SVM: Install and compile

- Create a new directory
  - $ mkdir svm_light
- Move svm_light.tar.gz into svm_light and decompress
  - $ tar xzf svm_light.tar.gz
- Compile
  - $ make
- Two executables will be created.
  - svm_learn (learning module)
  - svm_classify (classification module)

Learning Module

- svm_learn [options] example_file model_file
  - options: Refer help messages using “-?” option
  - example_file: Input file for training examples.
    - Format for classification mode
      - <Target> <Feature1>:<Value1> <F2>:<V2>…<Fn>:<Vn>
        - Target: +1 | -1 | 0
        - Feature: <integer>, Value: <float>
          - Feature/value pairs MUST be ordered by increasing feature number.
      - For example
        - -1 1:0.43 3:0.12 9284:0.2 --- Negative example
        - 1 1:0.1 10:0.45 --- Positive example
        - 0 1:0.34 5:0.13 189:0.5 --- Unknown example
  - model_file: Result of svm_learn is the model which is learned from the training examples.

Classification Module

- svm_classify [options] example_file model_file output_file
  - options: Refer help messages using “-?” option
  - example_file: Test examples in the same format as the training examples.
  - model_file: The model_file from svm_learn.
  - output_file
    - The result of svm_classify which has the predicted values.
    - The predicted values are result of the decision function for each examples.
    - The sign of the predicted value is the predicted class.
    - The zero indicates unknown

SVM 실행 예

- Example
  - http://download.joachims.org/svm_light/examples/example1.tar.gz

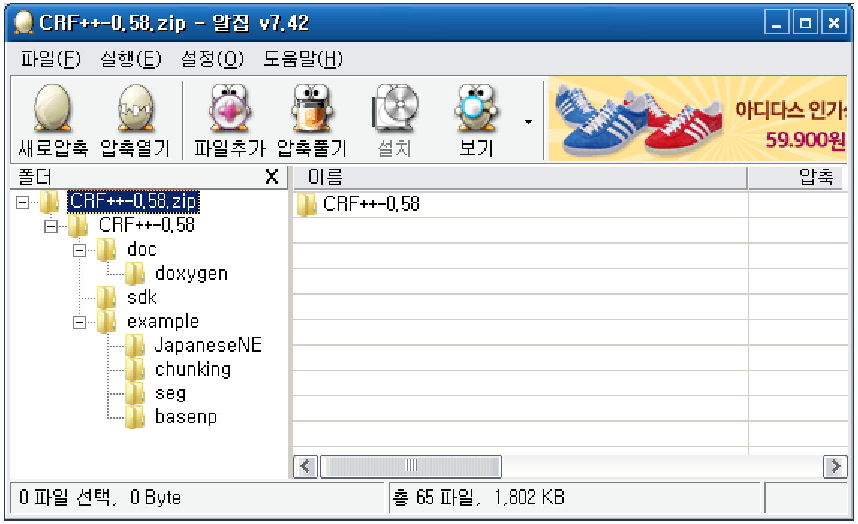
  - The task is to learn which Reuters articles are about "corporate acquisitions“.
  - 9947 features : Each feature corresponds to a word stem.
  - train.dat : 1000 positive and 1000 negative examples
  - test.dat : 600 test examples
  - words : A set of word stems. Features correspond to the line numbers. (9947 lines)

- 학습 모델 생성 및 실행
  $ svm_learn train.dat model
  $ svm_classify test.dat model predictions

CRF++

- http://crfpp.googlecode.com/svn/trunk/doc/index.html#download

- CRF++-0.58.tar.gz -- Source
- CRF++-0.58.zip
  - Binary for MS-Windows



CRF 통합 가능한 언어

- C++, Java, Python, Perl, Ruby 등

언어	설치 Directory	설명	비고
C++	CRF++-0.58/sdk	C++에서 CRF++라이브러리 연동 방법 제공	swig를 이용한 스크립트언어 C++ 라이브러리 인터페이스
JAVA	CRF++-0.58/java	JAVA 에서 CRF++라이브러리 연동 방법 제공	
Python	CRF++-0.58/python	Python 에서 CRF++라이브러리 연동 방법 제공	
Perl	CRF++-0.58/perl	Perl 에서 CRF++라이브러리 연동 방법 제공	
Ruby	CRF++-0.58/ruby	Ruby 에서 CRF++라이브러리 연동 방법 제공	

CRF++-0.58/example/basenp/

```
[taeseok@localhost CRF++-0.58]$ cd example/basenp/
exec.sh template test.data train.data
[taeseok@localhost python]$ ../../crf_learn -c 10.0 template train.data model
…
iter=33 terr=0.00000 serr=0.00000 act=32970 obj=19.70277 diff=0.00019
iter=34 terr=0.00000 serr=0.00000 act=32970 obj=19.70237 diff=0.00002
iter=35 terr=0.00000 serr=0.00000 act=32970 obj=19.70003 diff=0.00012
iter=36 terr=0.00000 serr=0.00000 act=32970 obj=19.69958 diff=0.00002
iter=37 terr=0.00000 serr=0.00000 act=32970 obj=19.69887 diff=0.00004
iter=38 terr=0.00000 serr=0.00000 act=32970 obj=19.69855 diff=0.00002

Done!0.15 s

[taeseok@localhost python]$ ../../crf_test  -m model test.data > output.txt
…
of	IN	O	O
Columbus	NNP	B	B
,	,	O	O
Ohio	NNP	B	B
,	,	O	O
grew	VBD	O	O
3.8	CD	B	B
%	NN	I	I
.	.	O	O

[taeseok@localhost python]$ ./conlleval.pl -d "\t" < output.txt
processed 19172 tokens with 5051 phrases; found: 4978 phrases; correct: 4285.
accuracy:  93.67%; precision:  86.08%; recall:  84.83%; FB1:  85.45
             : precision:  86.08%; recall:  84.83%; FB1:  85.45  4978
             : precision:  86.08%; recall:  84.83%; FB1:  85.45  4978
```

```
# Unigram
U00:%x[-2,0]
U01:%x[-1,0]
U02:%x[0,0]
U03:%x[1,0]
U04:%x[2,0]
U05:%x[-1,0]/%x[0,0]
U06:%x[0,0]/%x[1,0]

U10:%x[-2,1]
U11:%x[-1,1]
U12:%x[0,1]
U13:%x[1,1]
U14:%x[2,1]
U15:%x[-2,1]/%x[-1,1]
U16:%x[-1,1]/%x[0,1]
U17:%x[0,1]/%x[1,1]
U18:%x[1,1]/%x[2,1]

U20:%x[-2,1]/%x[-1,1]/%x[0,1]
U21:%x[-1,1]/%x[0,1]/%x[1,1]
U22:%x[0,1]/%x[1,1]/%x[2,1]

U23:%x[0,1]

# Bigram
B
```

http://www.cnts.ua.ac.be/conll2000/chunking/output.html

AI, ML, NN, and Deep Learning

- AI
  - 지식표현, game theory
  - NLP, Q&A, M.T., pattern recognition, expert system, etc
- Machine Learning
  - Decision tree, Neural Net, SVM, Naïve Bayes, Ada boost
- Deep Learning (Deep Neural Network)
  - Convolutional Neural Network (CNN)
  - Recurrent Neural Network (RNN)
  - Restricted Boltzmann Machine (RBM)

SMT and NMT

Example-based MT

- Basic idea: translate a sentence by using the closest match in parallel data.
- First proposed by Nagao (1981)
- Ex:
  - Training data:
    - w1 w2 w3 w4 → w1’ w2’ w3’ w4’
    - w5 w6 w7 → w5’ w6’ w7’
    - w8 w9 → w8’ w9’
  - Test sent:
    - w1 w2 w6 w7 w9 → w1’ w2’ w6’ w7’ w9’

- Types of EBMT:
  - Lexical (shallow)
  - Morphological / POS analysis
  - Parse-tree based (deep)

- Types of data required by EBMT systems:
  - Parallel text
  - Bilingual dictionary
  - Thesaurus for computing semantic similarity
  - Syntactic parser, dependency parser, etc.

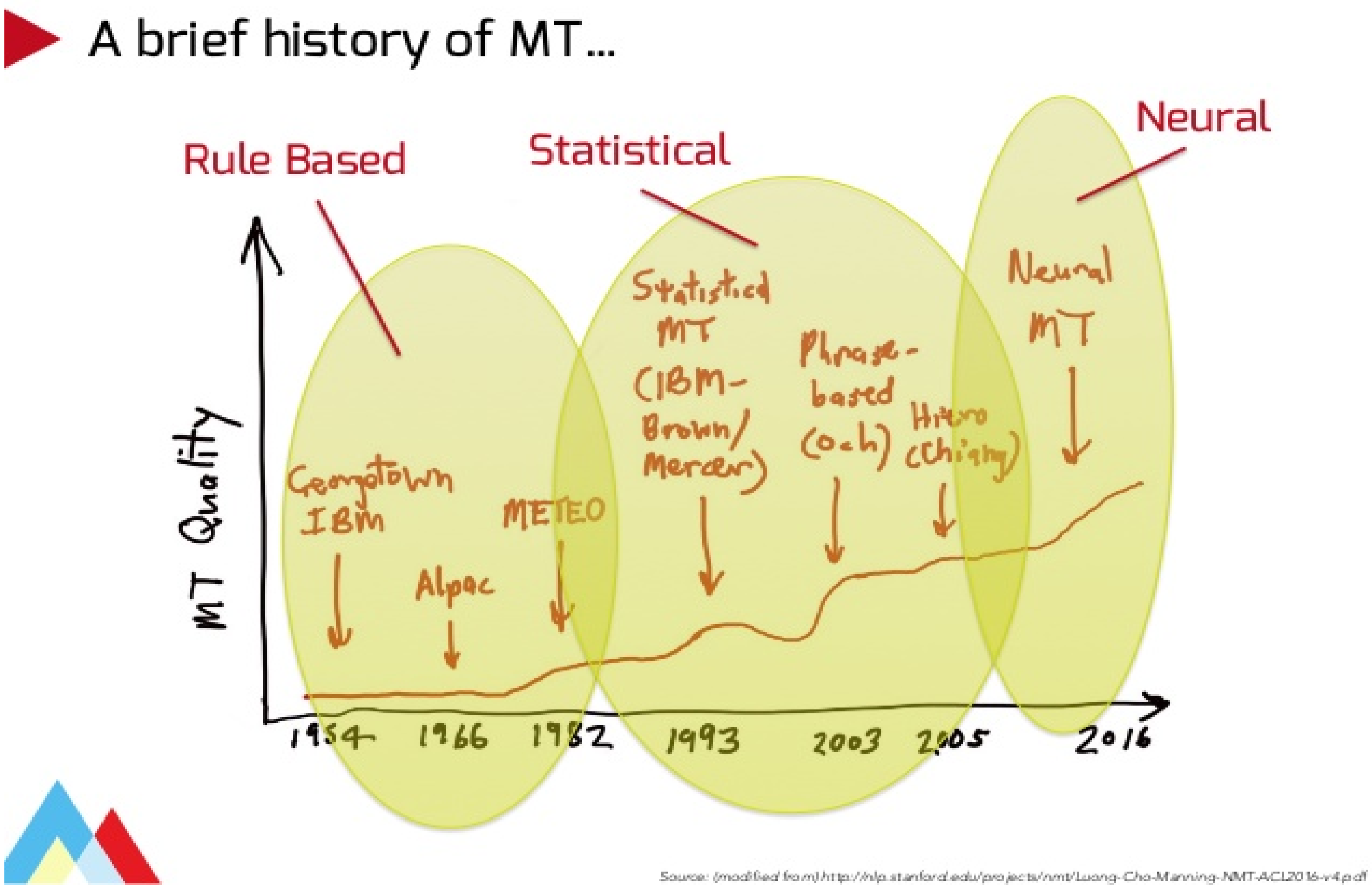
- Word alignment: using dictionary and heuristics
  ➔ exact match

- Generalization:
  - Clusters: dates, numbers, colors, shapes, etc.
  - Clusters can be built by hand or learned automatically.

- Ex:
  - Exact match: 12 players met in Paris last Tuesday ➔
    12 Spieler trafen sich letzen Dienstag in Paris

  - Templates: $num players met in $city $time ➔
    $num Spieler trafen sich $time in $city

Progress in M.T.



Statistical MT

- Basic idea: learn all the parameters from parallel data
- Major types: Word-based, Phrase-based

- Strengths:
  - Easy to build, and it requires no human knowledge
  - Good performance when a large amount of training data is available

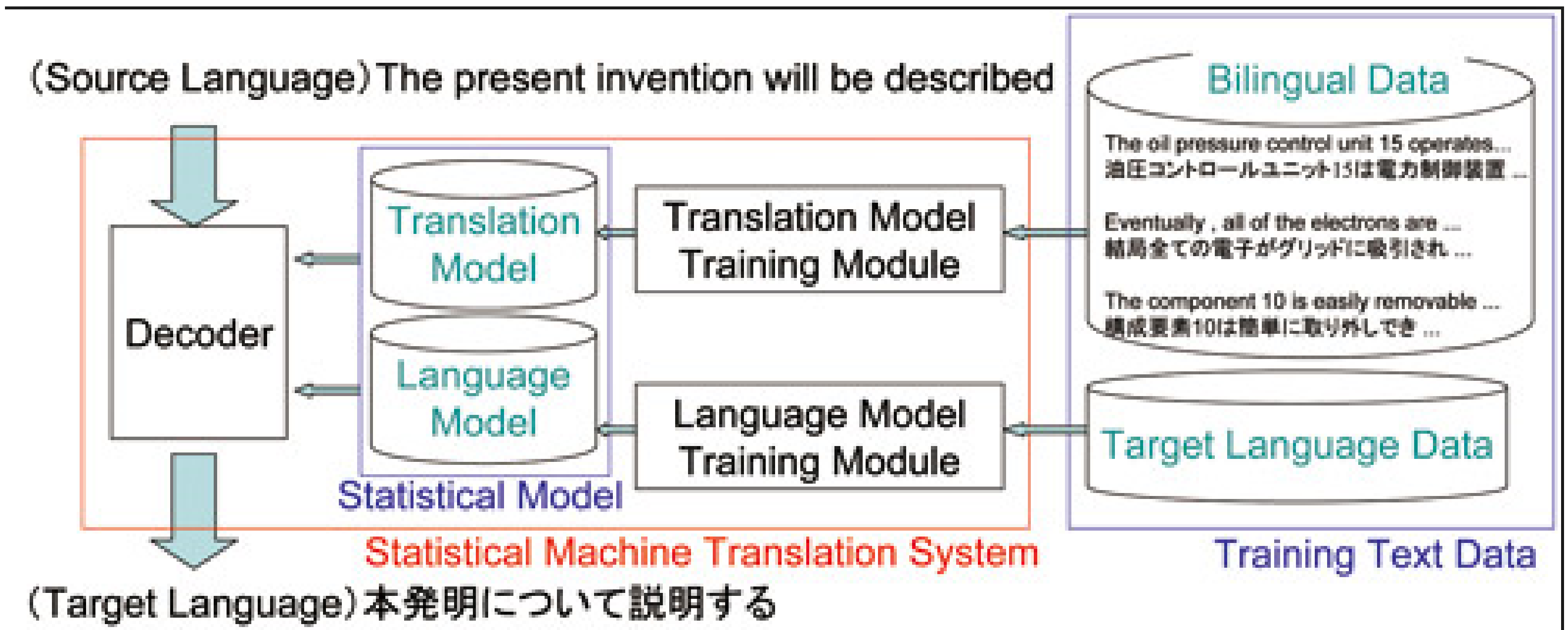
- Weaknesses:
  - How to express linguistic generalization?

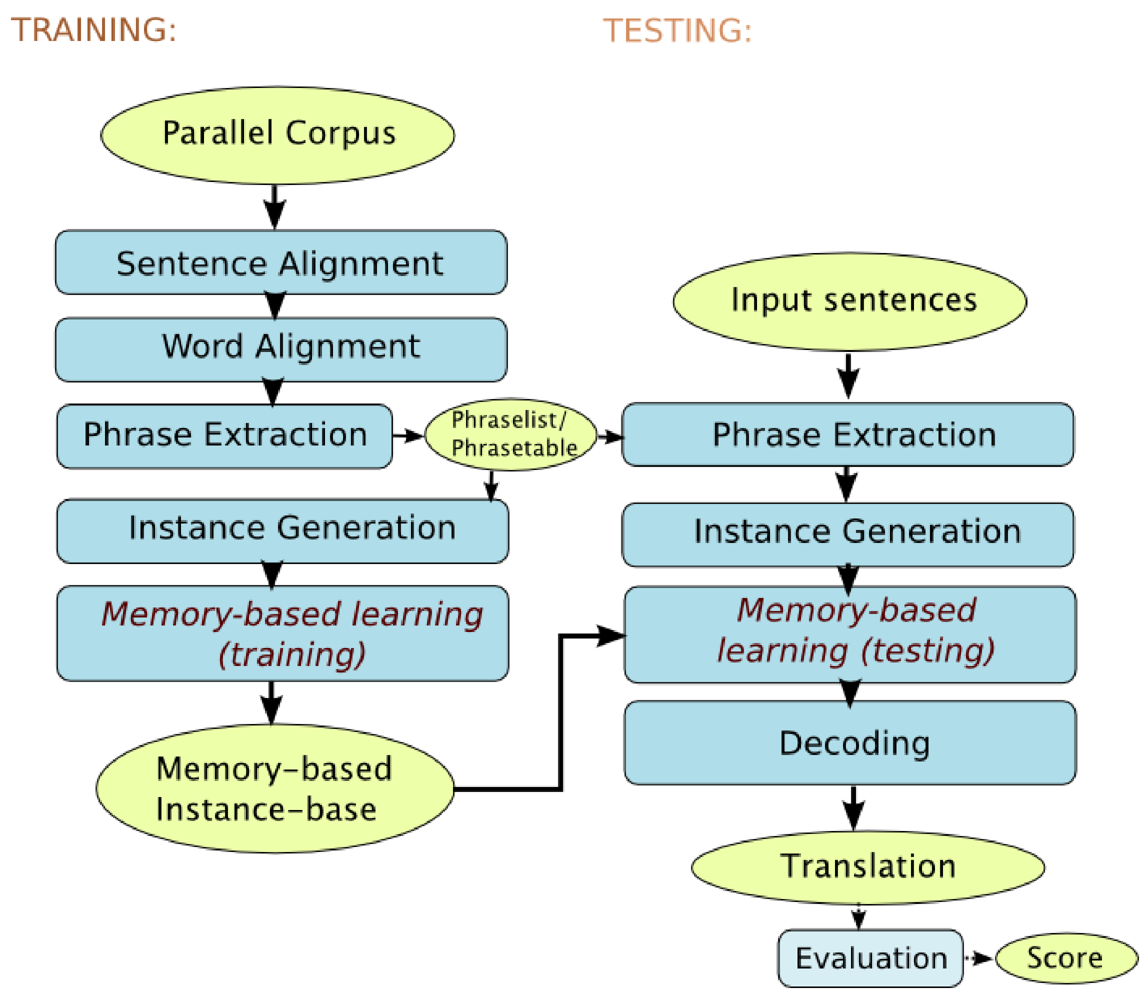
Hybrid MT

- Basic idea: combine different approaches

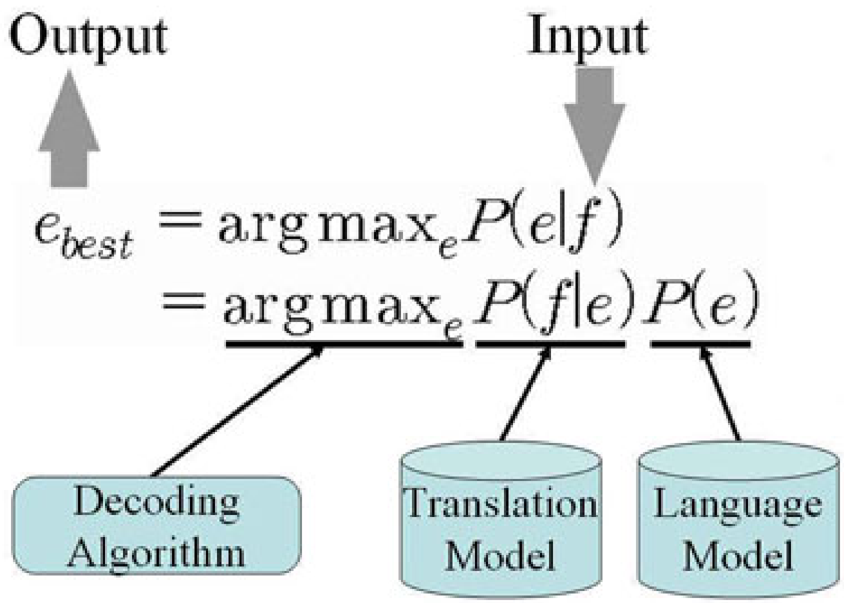
- Types of hybrid HT:
  - Borrowing concepts/methods:
    - SMT from EBMT: phrase-based SMT; Alignment templates
    - EBMT from SMT: automatically learned translation lexicon
    - Transfer-based from SMT: automatically learned translation lexicon, transfer rules; using LM
  - Using two MTs in a pipeline:
    - Using transfer-based MT as a preprocessor of SMT
  - Using multiple MTs in parallel, then adding a re-ranker

Statistical M.T. with Bilingual(Parallel) Corpus




TRAINING:
TESTING:
Parallel Corpus
Sentence Alignment
Word Alignment
Phrase Extraction
Phraselist/
Phrasetable
Instance Generation
Memory-based learning
(training)
Memory–based
Instance–base
Input sentences
Phrase Extraction
Instance Generation
Memory-based
learning (testing)
Decoding
Translation
Evaluation
Score

SMT Model


Output
Input
$e_{best} = \arg\max_e P(e|f)$
$= \underline{\arg\max_e}\ \underline{P(f|e)}\ \underline{P(e)}$
Decoding Algorithm
Translation Model
Language Model

Neural Machine Translation

- Demo -- http://104.131.78.120/

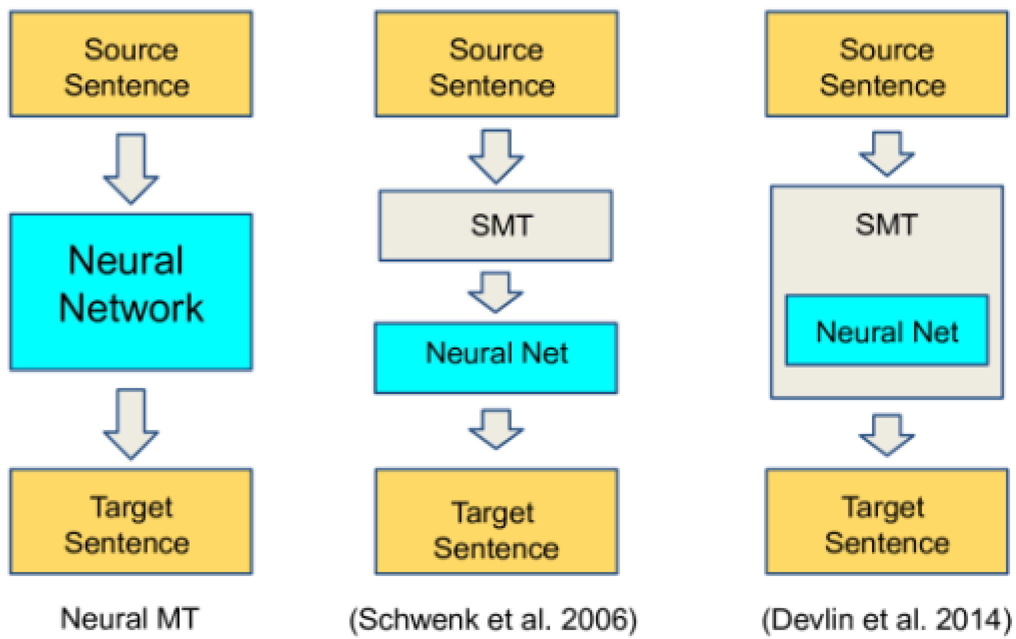


Figure 2. Graphical illustration of Neural MT, SMT+Reranking-by-NN and SMT-NN. From [Bahadanau et al., 2015] slides at ICLR 2015.

https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-with-gpus/

History Google Translator

- 2006, SYSTRAN
- 2007, SMT
- 2016, Google’s Multilingual Neural M.T.

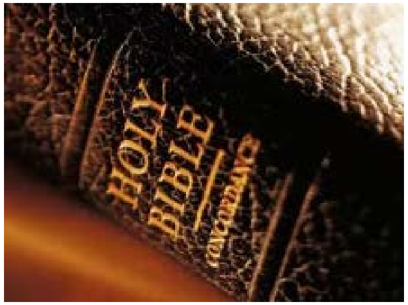
Traditional vs. Google Translate

- Traditional M.T. system
  - Break sentences into words and phrases
  - Translate each individually

- Google Translate, 2016/09
  - Neural translation system
  - Neural network to work on entire sentences at once
  - Multiple language combinations
    - Eng <-> Japanese & Eng <-> Korean → Kor <-> Japanese
    - By Cho Kyunghyun, New York Univ.

Learning the lingo: Google Translate

- gathers from across the internet
- community input
- the Bible for obscure languages



https://translate.google.com/

Babylon - Investments, M... | Google 번역

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)

국민대 KMU-NLP 웹메일 NAVER 지도 Daum Google Gmail Google 지도 KB국민 우리

영어 한국어 독일어 언어 감지 | 한국어 영어 일본어 | 번역하기

Imprisoned American Student, 22, Sent Home From North Korea Amid Reports He's in a Coma
An American student who hasn't been seen since North Korea convicted him of crimes against the country is reportedly on his way back home, but he's believed to be in troubling condition.
Otto Warmbier made international headlines in 2016 after North Korea claimed he attempted to steal a propaganda poster led to a sentence of 15 years hard labor. The 22-year-old hasn't been seen by American representatives in Pyongyang since.
Watch: Friends Shocked After 'Well-Liked' American Student is Arrested in North Korea for Alleged 'Hostile Act'
But on Tuesday, the already strange tale took some even stranger turns when it was reported that the Cincinnati native is on his way back to Ohio in a coma, one his parents have been told Warmbier has been in for nearly his entire incarceration.
Pyongyang officials have reportedly claimed Warmbier contracted botulism shortly after the conclusion of the hour-long trial in which he was found guilty of "hostile acts against the state."
The Warmbiers are told he was subsequently "given a sleeping pill, from which he never woke up," they told The Washington Post.
Swedish diplomats who represent American interests in the Hermit Kingdom, with which the U.S. has no ties, say they haven't been given access to Warmbier since the trial and there is no way to know whether North Korea's account is true.
Warmbier is being transported back via Japan, where State Department officials were slated to meet him for the journey back to Ohio.
In what an official has called a "bizarre coincidence," that journey is happening the same day that retired NBA star Dennis Rodman began his fifth high-profile visit to Pyongyang.
Weirder still, The Washington Post reports that the official believes the Basketball Hall of Famer may be visiting as part of an attempt by Pyongyang to distract from Warmbier's condition.
Either way, more will likely be known about Warmbier's physical state once he's back on American soil Tuesday evening.
Watch: Dennis Rodman Charged With Hit-and-Run After Allegedly Driving on Wrong Side of Freeway

미국계 미국인 학생, 22 명, 코마 상태의 보고서 가운데 북한으로부터 집을 보냈다.
북한이 범죄 사실로 유죄 판결을받은 이래로 본 적이없는 미국인 학생은 집으로 돌아 오는 중이라고 전해지고 있지만, 그는 곤경에 처한 것으로 여겨진다.
오토 와임 비어 (Otto Warmbier)는 북한이 15 년의 고된 노동을 선고 한 선전 포스터를 훔치려 고 주장하면서 2016 년 국제 표제를 만들었다. 22 세의 나이로 평양에있는 미국 대표들은 본 적이 없다.
조심 : '잘난 체'미국 학생이 북한에서 적대적인 '적대 행위'로 체포 된 후 친구들이 충격을 받았다.
그러나 화요일에, 이미 이상한 이야기는 신시내티 출신이 혼수 상태로 오하이오로 돌아 오는 중이라고 보도되었을 때, 그의 부모님은 웜비어가 거의 투옥되었다고 들었다.
북한 당국자들은 1 시간 동안의 재판이 끝난 직후에 워머 보어가 보툴리누스 중독에 걸렸다 고 주장하며, "국가에 대한 적대 행위"에 대한 유죄 판결을 받았다고 보도했다.
워머 비어 (Warmbiers)는 그가 연속적으로 "절대로 깨어나지 않는 수면제를 받았다"고 말했다.
미국이 연계되어 있지 않은 은둔자 왕국에 대한 미국의 이익을 대표하는 스웨덴 외교관은 재판이 있은 후 와임 비어 (Warmbier)에 대한 접근권이 주어지지 않았으며 북한의 주장이 사실인지 여부를 알 수있는 방법이 없다고 말한다.
Warmbier는 미 국무부 당국자가 오하이오로 돌아 가기위한 여행을 위해 그를 만나게 될 일본을 통해 다시 수송되고있다.
한 관리가 "기묘한 우연의 일치"라고 말한 것으로, 퇴역 한 NBA 스타 데니스로드 만 (Dennis Rodman)이 평양으로 다섯 번째로 유명한 방문을 시작한 그 날 여행이 일어나고있다.
워너 포스트는 워싱턴 포스트지가 농무부의 명예의 전당이 웜비어의 상태를 혼란에 빠뜨리려는 시도의 일환으로 방문했다고 믿고 있다고 보도했다.
어느 쪽이든 화요일 저녁 미국 땅에 돌아온 후에는 웜비어의 신체 상태에 대해 더 많이 알게 될 것입니다.
조심 : 데니스 로드맨 (Dennis Rodman)이 고속도로의 잘못된 쪽을 운전하다가 히트 앤 앤드 런 (Hit-and-Run)으로 청구 됨
오하이오 주 상원 의원 롭 포트먼 (Ham Portman)은 뉴스를 확인하는 성명서에서 처음에는 웜비어 (Warmbier)의 투옥에 대해 비난했다.
포트먼 장관은 성명을 통해 "북한은 혐오스러운 행동으로 보편적으로 비난 받아야한다"고 밝혔다. 그는 "북한이 처음부터 석방 돼야한다"며 "북한이 1 년 넘게 통고 또는 영사를받지 않고 투옥하면 인권과 존엄을 인정하지 않는 것이 최선의 사례"라고 덧붙였다.

수정 제안하기

SYSTRANet – Onl

도움말(H)

지도 Daum Google Gmail Google 지도

itor

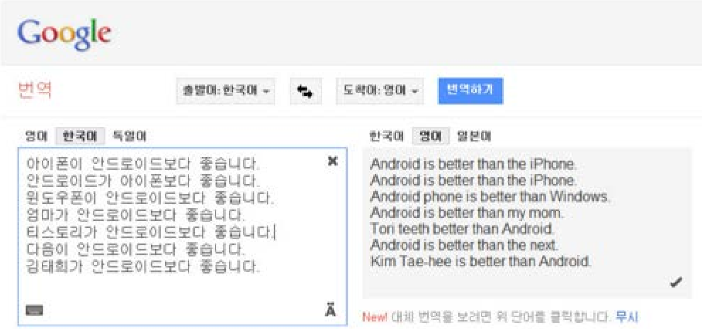
Dictionary | My Dictionary | Register for free

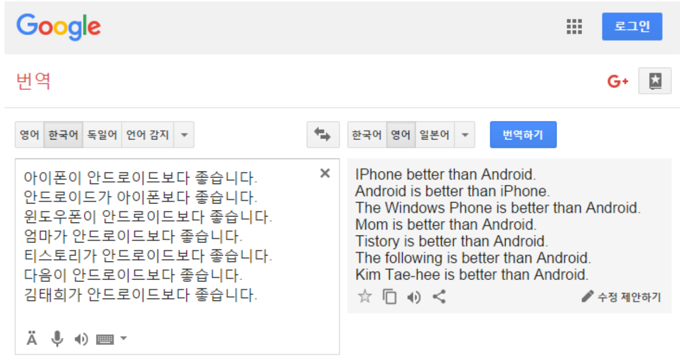
Translate | Options:

투옥한 미국 학생, 보고의 속에 북한에게서 집으로 보내진 22 그는 코마에서 이다
북한이 그의 방법 뒤 가정에 국가에 대하여 범죄에 대하여 그를 소문에 의하면 있을 유죄를 입증할 후 부터 보이지 않은 미국 학생, 그러나 그는 조건 고생에서 이기 위하여 생각했다.
북한이 그를 중노동 15 년의 문장에 지도된 주의 포스터를 훔친 것을 시도한 요구한 후에 오토 Warmbier는 2016년에 국제적인 표제를 만들었다. 22세는 평양에 있는 미국 대표자에 의해 그 후 보이지 않았다.
시계: '잘 좋아한' 미국 학생 후에 충격 받은 친구는 추정된 '적대하는 행위'를 위한 북한에서 검거된다
그러나 화요일에, 이미 이상한 이야기는 신시내티 출신은 코마에 있는 오하이오 등을 맞댄 그의 방법에 다는 것을, Warmbier는 그의 전체 투옥을 위해 거의 안으로 이었다는 것을 그의 부모가 말된 1개를 보고될 때 몇몇 더 이상한 회전을 조차 가지고 갔다.
평양 관리는 소문에 의하면 그가 "국가에 대하여 적대하는 행위의 유죄이라고 " 선고받은 시

빅데이터 활용 예: 구글 번역

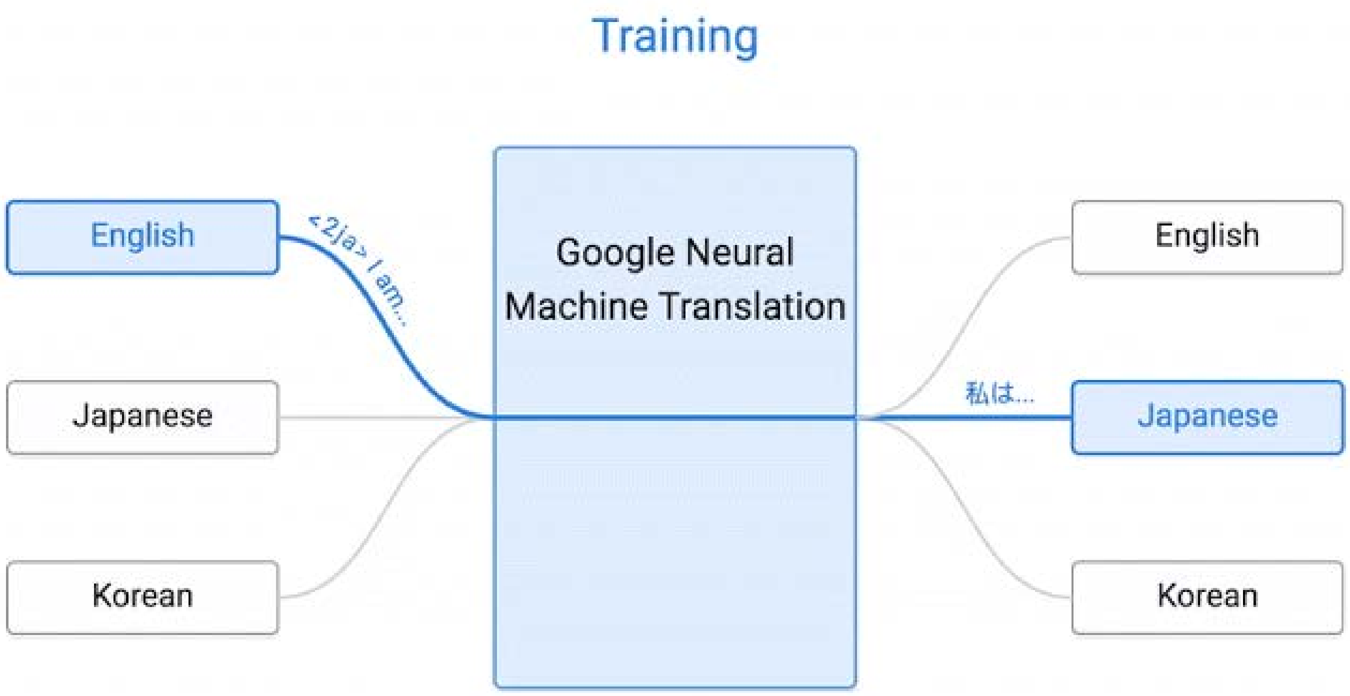
- 기존의 기계번역 방식
  - 변환(transfer) 방식과 피봇(pivot) 방식의 자동 번역 기법
  - 컴퓨터가 명사, 형용사, 동사 등 단어와 어문의 문법적 구조를 인식하여 번역하는 방식
- 구글이 제공하는 자동 번역 서비스인 구글 번역의 특징
  - 통계적 방식: 빅데이터를 활용하는 방법으로 구현
  - 수억 건의 문장과 번역문을 데이터베이스화
  - 번역시 유사한 문장과 어구를 기존에 축적된 데이터를 바탕으로 추론
  - 구글은 수억 건의 자료를 활용하여 전 세계 58개 언어 간의 자동번역 프로그램 개발에 성공
- 데이터 양의 측면에서의 엄청난 차이가 자동 번역 프로그램의 번역의 질과 정확도에 영향을 미침



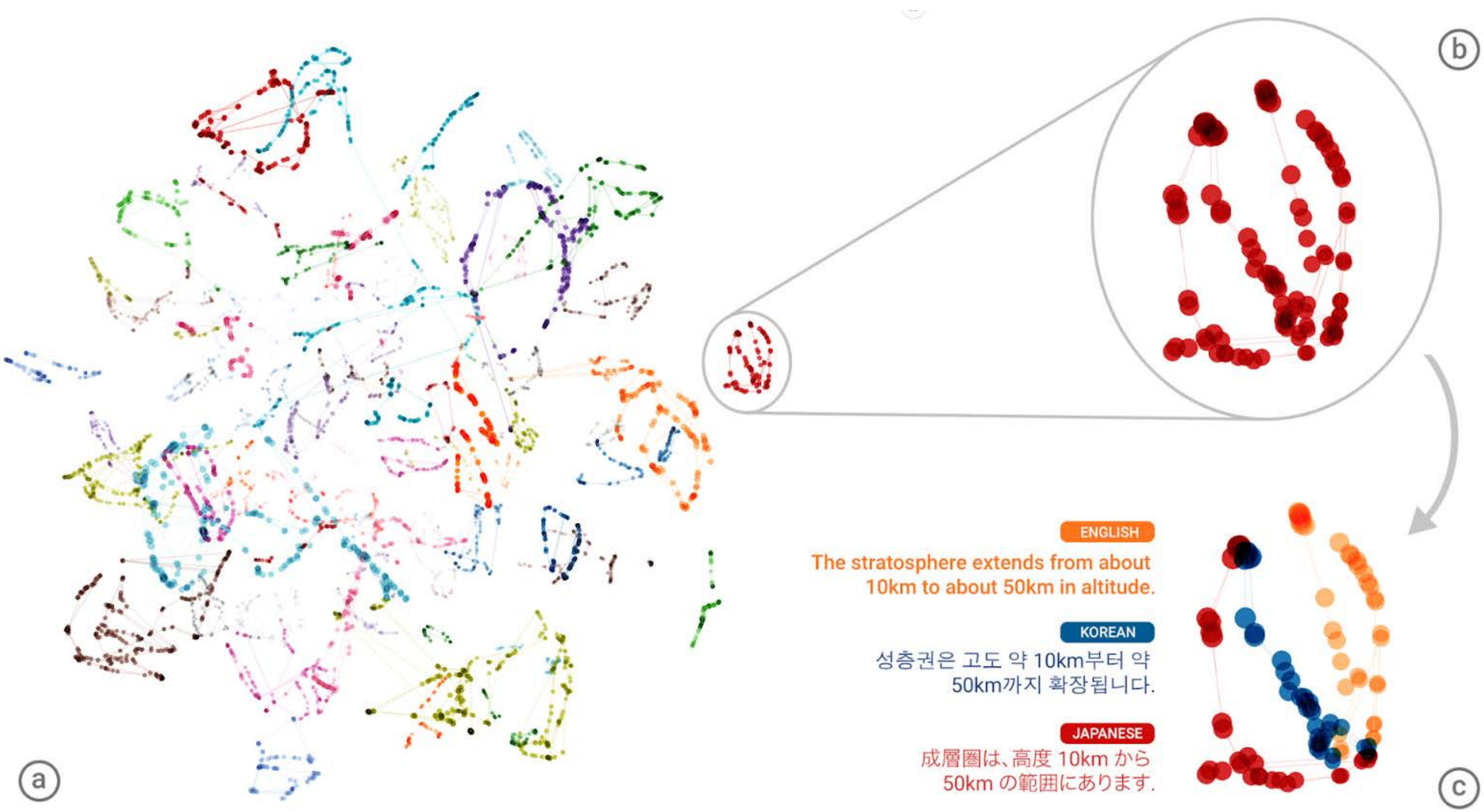


GNMT: Google's Multilingual Neural Machine Translation System

- Zero-Shot Translation



Zero-Shot Translation



- Part (a) shows an overall geometry of these translations.
    - The points in this view are colored by the meaning; a sentence translated from English to Korean with the same meaning as a sentence translated from Japanese to English share the same color.
    - From this view we can see distinct groupings of points, each with their own color.
- Part (b) zooms in to one of the groups.
- Part (c) colors by the source language.
    - Within a single group, we see a sentence with the same meaning but from three different languages.
    - This means the network must be encoding something about the semantics of the sentence rather than simply memorizing phrase-to-phrase translations.
    - We interpret this as a sign of existence of an interlingua in the network.

Google's Neural Machine Translation System: Bridging the Ga between Human and Machine Translation

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi
yonghui,schuster,zhifengc,qvl,mnorouzi@google.com

Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean

Abstract

Neural Machine Translation (NMT) is an end-to-end learning approach for automated translation, with the potential to overcome many of the weaknesses of conventional phrase-based translation systems. Unfortunately, NMT systems are known to be computationally expensive both in training and in translation inference – sometimes prohibitively so in the case of very large data sets and large models. Several authors have also charged that NMT systems lack robustness, particularly when input sentences contain rare words. These issues have hindered NMT's use in practical deployments and services, where both accuracy and speed are essential. In this work, we present GNMT, Google's Neural Machine Translation system, which attempts to address many of these issues. Our model consists of a deep LSTM network with 8 encoder and 8 decoder layers using residual connections as well as attention connections from the decoder network to the encoder. To improve parallelism and therefore decrease training time, our attention mechanism connects the bottom layer of the decoder to the top layer of the encoder. To accelerate the final translation speed, we employ low-precision arithmetic during inference computations. To improve handling of rare words, we divide words into a limited set of common sub-word units ("wordpieces") for both input and output. This method provides a good balance between the flexibility of "character"-delimited models and the efficiency of "word"-delimited models, naturally handles translation of rare words, and ultimately improves the overall accuracy of the system. Our beam search technique employs a length-normalization procedure and uses a coverage penalty, which encourages generation of an output sentence that is most likely to cover all the words in the source sentence. To directly optimize the translation BLEU scores, we consider refining the models by using reinforcement learning, but we found that the improvement in the BLEU scores did not reflect in the human evaluation. On the WMT'14 English-to-French and English-to-German benchmarks, GNMT achieves competitive results to state-of-the-art. Using a human side-by-side evaluation on a set of isolated simple sentences, it reduces translation errors by an average of 60% compared to Google's phrase-based production system.

특집원고

Neural Machine Translation 기반의 영어-일본어 자동번역

강원대학교 ∥ 이창기
네이버 ∥ 김준석·이형규·이재송

1. 서 론

최근의 기계번역 연구에서 Neural Machine Translation (NMT) 모델이 큰 각광을 받고 있다. 최근까지 기계번역에 신경망을 적용하는 방식은re-ranking 방식이 주로 연구되었고, 최근에는 end-to-end 방식의 신경망 구조를 사용하는 NMT 모델이 영어-프랑스와 같이 어순이 유사한 언어 쌍에서 좋은 성능을 보였다.

본 논문에서는 전통적인 방식의 SMT 방식인 구 기반(Phrase-based) 모델과 계층적 구 기반(Hierarchical Phrase-based) 모델, 그리고 구문 기반(Syntax-based) 모델을 소개한다. 또한 최근에 각광받고 있는 NMT 모델에 대해 설명하고 이를 어순이 상이한 영어-일본어 기계번역에 적용한다. 실험을 통해 NMT 모델이 기존의 구 기반 모델과 계층적 구 기반 모델보다 성능이 우수하고, 구문분석을 사용하는 Syntax-based 모델과 성능이 유사함을 보인다.

2장에서는 전통적인 방식의 SMT 모델에 대해서 소개하고, 3장에서는 NMT 모델에 대해서 설명하고 NMT 모델의 장단점에 대해서 알아본다. 4장에서는 기존의 SMT 모델과 NMT 모델을 어순이 상이한 영어-일본어 기계번역에 적용한 결과를 설명한다.

기반으로 최적의 번역 열을 찾은 후에 출력 언어(target language) 문장에 맞게 생성하면 번역이 완료된다.

그림 2는 구(phrase) 기반의 SMT의 기본 수식을 보여준다. f_i는 입력 언어의 구(phrase)를 e_i는 출력 언어의 구를 의미한다. $P(f_i|e_i)$는 e_i가 f_i로 번역될 확률 값이고, $P(e_i)$은 출력 언어가 나타날 확률인 언어 모델 값이다. 수식에서 양변에 log를 취하는 log-linear 모델로 변환하면 그림 2의 아래와 같은 가중치-합(weighted-sum) 형태의 수식이 된다. $h_m(e, f)$은 번역 모델과 언어 모델 같은 feature 함수가 되고, λ_m은 해당 feature 함수의 가중치를 의미한다. 가중치는 기계번역에서 가장 많이 사용하는 척도인 BLEU[1]값을 최대화 시키는 파라미터 최적화 방식 MERT[2]를 사용하여 그 값이 결정된다. 따라서 전통적인 방식의 SMT는 결국 좋은 feature 함수를 발굴하는 것이 가장 중요하다. 이에 따라, 번역 모델, 언어 모델 외에 다양한 feature 함수를 만들어내고 기존 모델에 추가하여 실험을 통해 번역 품질 높이는 많은 연구들이 진행되었다.

제27회 한글 및 한국어 정보처리 학술대회 논문집 (2015년)

문자 단위의 Neural Machine Translation

이창기°, 김준석, 이형규, 이재송
강원대학교°, 네이버 랩스
leeck@kangwon.ac.kr, {jun.seok, hg.lee, jaesong.lee}@navercorp.com

Character-Level Neural Machine Translation

Changki Lee°, Junseok Kim, Hyoung-Gyu Lee, Jaesong Lee
Kangwon National University°, NAVER LABS

요 약

Neural Machine Translation (NMT) 모델은 단일 신경망 구조만을 사용하는 End-to-end 방식의 기계번역 모델로, 기존의 Statistical Machine Translation (SMT) 모델에 비해서 높은 성능을 보이고, Feature Engineering이 필요 없으며, 번역 모델 및 언어 모델의 역할을 단일 신경망에서 수행하여 디코더의 구조가 간단하다는 장점이 있다. 그러나 NMT 모델은 출력 언어 사전(Target Vocabulary)의 크기에 비례해서 학습 및 디코딩의 속도가 느려지기 때문에 출력 언어 사전의 크기에 제한을 갖는다는 단점이 있다. 본 논문에서는 NMT 모델의 출력 언어 사전의 크기 제한 문제를 해결하기 위해서, 입력 언어는 단어 단위로 읽고(Encoding) 출력 언어를 문자(Character) 단위로 생성(Decoding)하는 방법을 제안한다. 출력 언어를 문자 단위로 생성하게 되면 NMT 모델의 출력 언어 사전에 모든 문자를 포함할 수 있게 되어 출력 언어의 Out-of-vocabulary(OOV) 문제가 사라지고 출력 언어의 사전 크기가 줄어들어 학습 및 디코딩 속도가 빨라지게 된다. 실험 결과, 본 논문에서 제안한 방법이 영어-일본어 및 한국어-일본어 기계번역에서 기존의 단어 단위의 NMT 모델보다 우수한 성능을 보였다.

References

- Austermühl, Frank (2001) Electronic Tools For Translators
- http://www.essex.ac.uk/linguistics/clmt/MTbook/

  an introductory guide to MT by D.J.Arnold (1994)
- Free-to-use machine translation on the web:
  - http://www.translatorsbase.com/ (Free human translation service)
  - http://www.freetranslation.com/
  - http://www.tranexp.com:2000/InterTran?from=fre
  - http://www.systransoft.com/
  - http://www.systranet.com/ (the Systran site)
  - http://www.babylon.com/
  - http://www.reverso.net/textonly/default_ie.asp
  - http://translate. google.com/

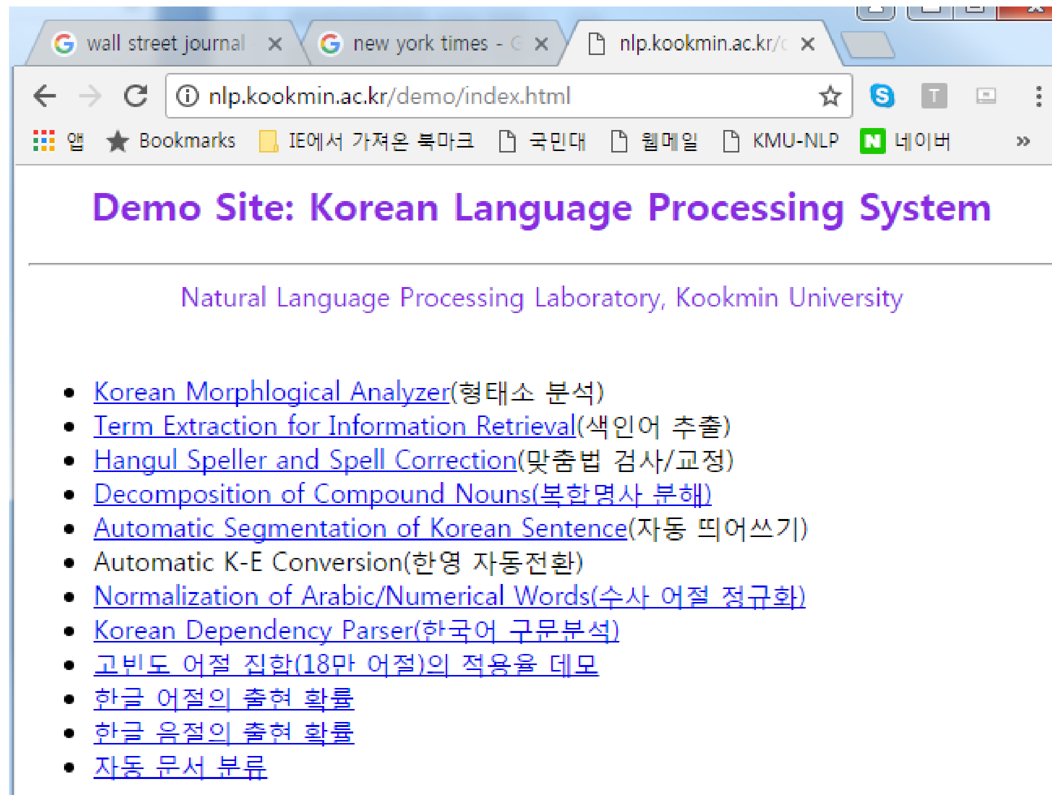
마지막으로…

http://nlp.kookmin.ac.kr/
http://cafe.daum.net/nlpk

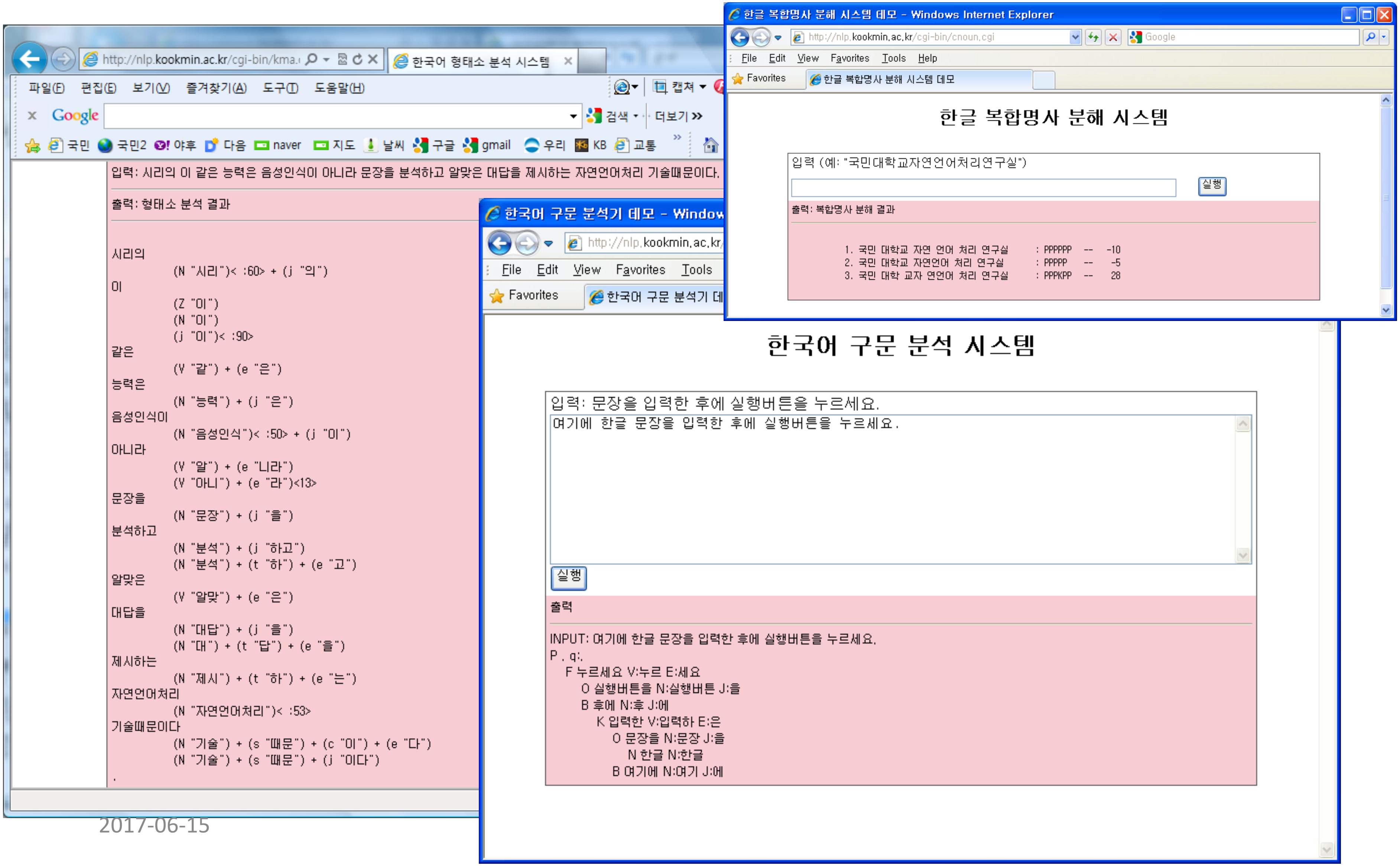
- 한국어 형태소 분석
- 구문 분석
- 색인어 추출 및 가중치 계산

- 복합명사 분해
- 맞춤법 검사 및 교정

- 자동 문서 분류
- 자동 띄어쓰기 등

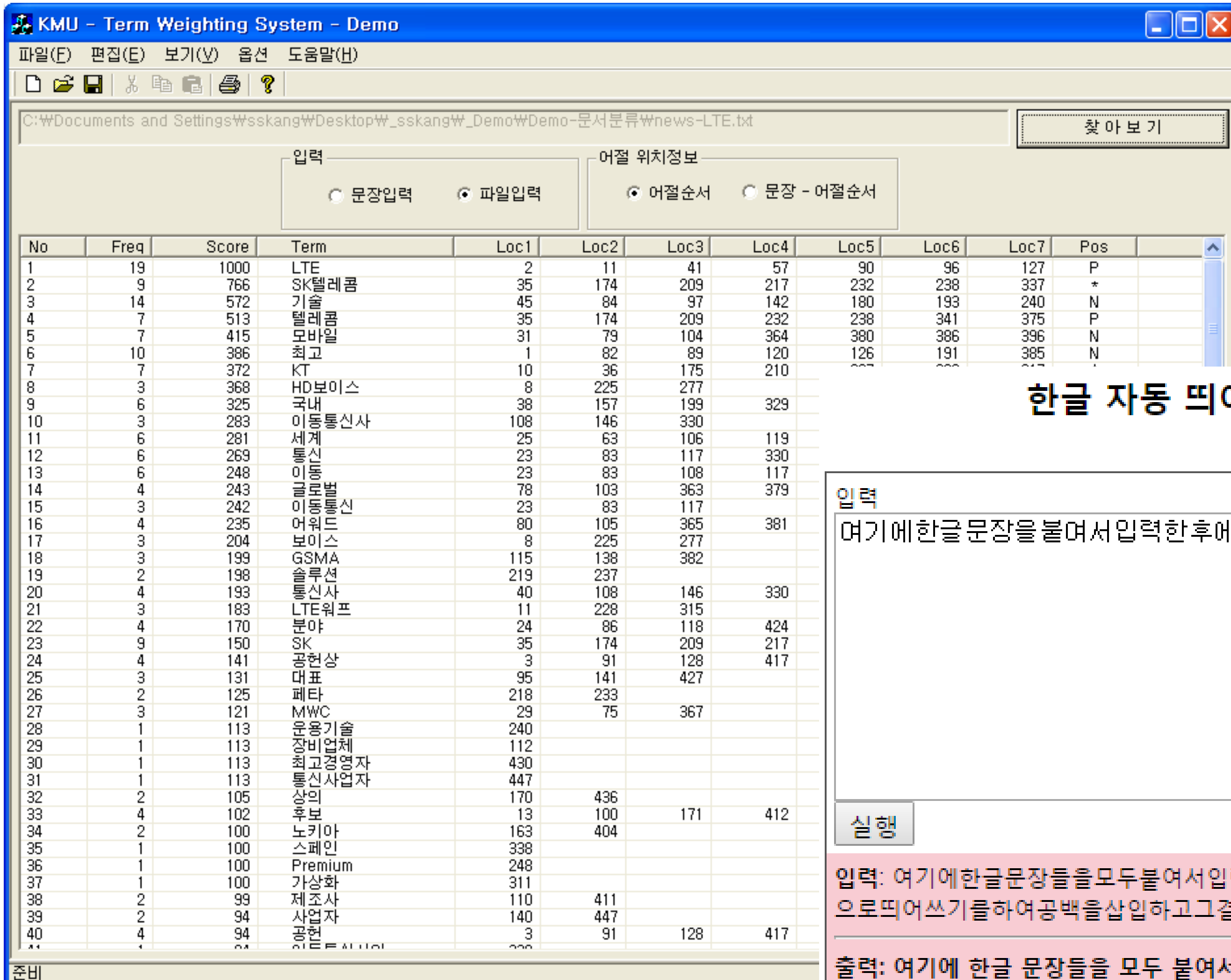


형태소 분석과 구문분석

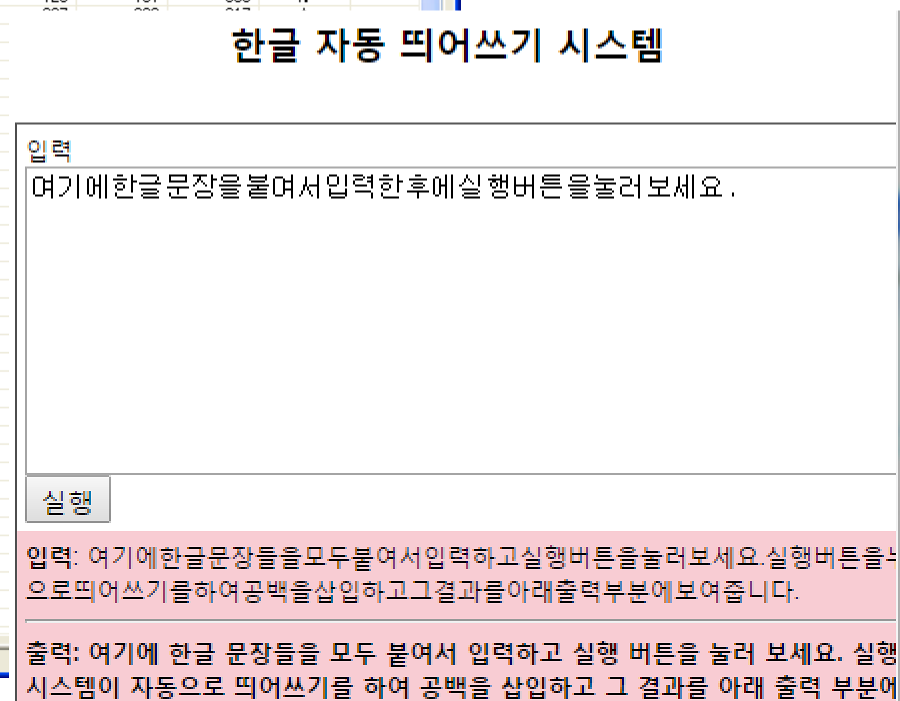


http://nlp.kookmin.ac.kr/cgi-bin/kma.
한국어 형태소 분석 시스템
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
입력: 시리의 이 같은 능력은 음성인식이 아니라 문장을 분석하고 알맞은 대답을 제시하는 자연언어처리 기술때문이다.
출력: 형태소 분석 결과
시리의
(N "시리")< :60> + (j "의")
이
(Z "이")
(N "이")
(j "이")< :90>
같은
(V "같") + (e "은")
능력은
(N "능력") + (j "은")
음성인식이
(N "음성인식")< :50> + (j "이")
아니라
(V "알") + (e "니라")
(V "아니") + (e "라")<13>
문장을
(N "문장") + (j "을")
분석하고
(N "분석") + (j "하고")
(N "분석") + (t "하") + (e "고")
알맞은
(V "알맞") + (e "은")
대답을
(N "대답") + (j "을")
(N "대") + (t "답") + (e "을")
제시하는
(N "제시") + (t "하") + (e "는")
자연언어처리
(N "자연언어처리")< :53>
기술때문이다
(N "기술") + (s "때문") + (c "이") + (e "다")
(N "기술") + (s "때문") + (j "이다")
한국어 구문 분석기 데모 - Window
http://nlp.kookmin.ac.kr
File Edit View Favorites Tools
Favorites
한국어 구문 분석기 데
한국어 구문 분석 시스템
입력: 문장을 입력한 후에 실행버튼을 누르세요.
여기에 한글 문장을 입력한 후에 실행버튼을 누르세요.
실행
출력
INPUT: 여기에 한글 문장을 입력한 후에 실행버튼을 누르세요.
P . q:.
F 누르세요 V:누르 E:세요
O 실행버튼을 N:실행버튼 J:을
B 후에 N:후 J:에
K 입력한 V:입력하 E:은
O 문장을 N:문장 J:을
N 한글 N:한글
B 여기에 N:여기 J:에
한글 복합명사 분해 시스템 데모 - Windows Internet Explorer
http://nlp.kookmin.ac.kr/cgi-bin/cnoun.cgi
File Edit View Favorites Tools Help
Favorites
한글 복합명사 분해 시스템 데모
한글 복합명사 분해 시스템
입력 (예: "국민대학교자연언어처리연구실")
실행
출력: 복합명사 분해 결과
1. 국민 대학교 자연 언어 처리 연구실 : PPPPPP -- -10
2. 국민 대학교 자연언어 처리 연구실 : PPPPP -- -5
3. 국민 대학 교자 연언어 처리 연구실 : PPPKPP -- 28

문서에서 키워드 추출



No	Freq	Score	Term	Loc1	Loc2	Loc3	Loc4	Loc5	Loc6	Loc7	Pos
1	19	1000	LTE	2	11	41	57	90	96	127	P
2	9	766	SK텔레콤	35	174	209	217	232	238	337	*
3	14	572	기술	45	84	97	142	180	193	240	N
4	7	513	텔레콤	35	174	209	232	238	341	375	P
5	7	415	모바일	31	79	104	364	380	386	396	N
6	10	386	최고	1	82	89	120	126	191	385	N
7	7	372	KT	10	36	175	210				
8	3	368	HD보이스	8	225	277					
9	6	325	국내	38	157	199	329				
10	3	283	이동통신사	108	146	330					
11	6	281	세계	25	63	106	119				
12	6	269	통신	23	83	117	330				
13	6	248	이동	23	83	108	117				
14	4	243	글로벌	78	103	363	379				
15	3	242	이동통신	23	83	117					
16	4	235	어워드	80	105	365	381				
17	3	204	보이스	8	225	277					
18	3	199	GSMA	115	138	382					
19	2	198	솔루션	219	237						
20	4	193	통신사	40	108	146	330				
21	3	183	LTE워프	11	228	315					
22	4	170	분야	24	86	118	424				
23	9	150	SK	35	174	209	217				
24	4	141	공헌상	3	91	128	417				
25	3	131	대표	95	141	427					
26	2	125	페타	218	233						
27	3	121	MWC	29	75	367					
28	1	113	운용기술	240							
29	1	113	장비업체	112							
30	1	113	최고경영자	430							
31	1	113	통신사업자	447							
32	2	105	상의	170	436						
33	4	102	후보	13	100	171	412				
34	2	100	노키아	163	404						
35	1	100	스페인	338							
36	1	100	Premium	248							
37	1	100	가상화	311							
38	2	99	제조사	110	411						
39	2	94	사업자	140	447						
40	4	94	공헌	3	91	128	417				



감사합니다!

sskang@kookmin.ac.kr