How can we capture multiword expressions?

Seongmin Mun[1], Guillaume Desagulier[2], Kyungwon Lee[3]

[1] Lifemedia Interdisciplinary Program, Ajou University, South Korea
[1] UMR 7114 MoDyCo - CNRS, University Paris Nanterre, France
[2] UMR 7114 MoDyCo - University Paris 8, CNRS, University Nanterre
[3] Department of Digital Media, Ajou University, South Korea

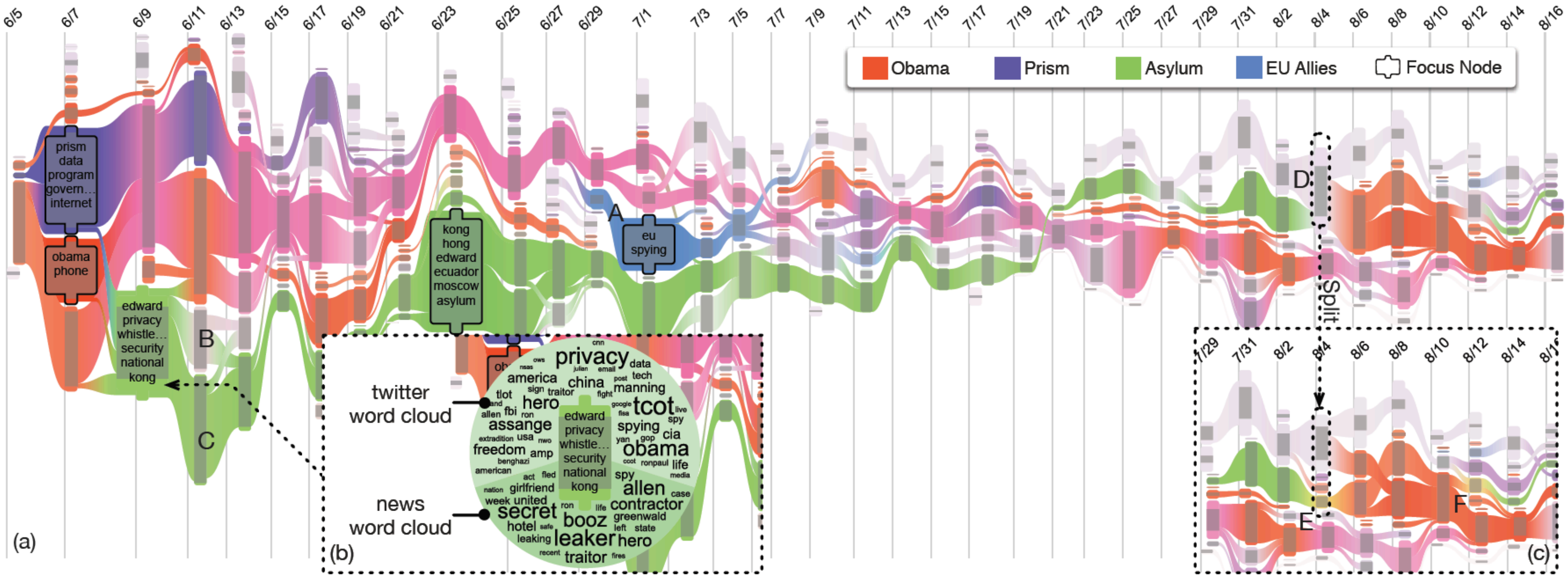
Université Paris Nanterre AJOU UNIVERSITY

Introduction

Words in a text corpus include features and information.

Analyzing these words can improve a user’s understanding of the corpus.

Previous studies



WEIWEI CUI SHIXIA LIU Z. W. H. W.: How hierarchical topics evolve in large text corpora. In IEEE Tran sactions on Visualization and Computer Graphics (2014), vol. 20, pp. 2281–2290.

Research background and purpose

Words can be broadly divided into two categories.

Research background and purpose

“With profound gratitude and great humility, I accept your nomination for the presidency of the United States.”

Research background and purpose

“With profound ***gratitude*** and great humility, I accept your nomination for the presidency of the United States.”

Gratitude ➡ meaning that can be expressed in one word

Research background and purpose

“With profound gratitude and great humility, I accept your nomination for the presidency of the ***United States***.”

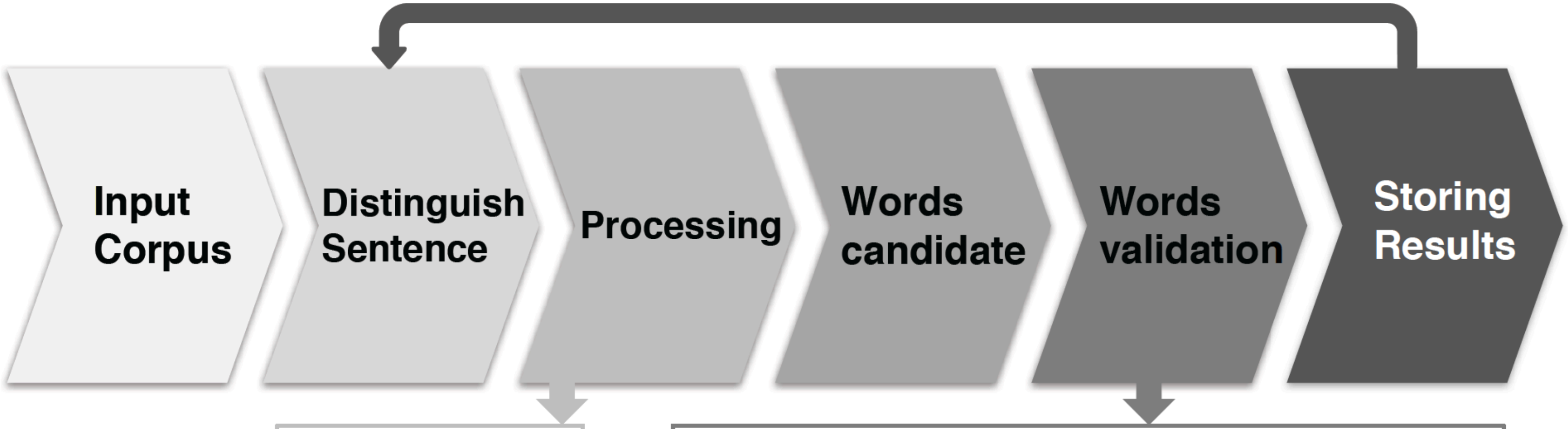
United States ➡ meaning must be described using a combination of words.

Research background and purpose

How can we capture multiword expressions?

To this aim, we designed an algorithm.

Data processing



Input Corpus

Distinguish Sentence

Processing

Words candidate

Words validation

Storing Results

Processing

- N-grams
- Dependency tag
- POS tagging

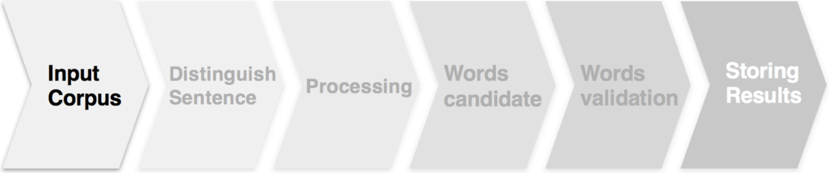
Pre-processing

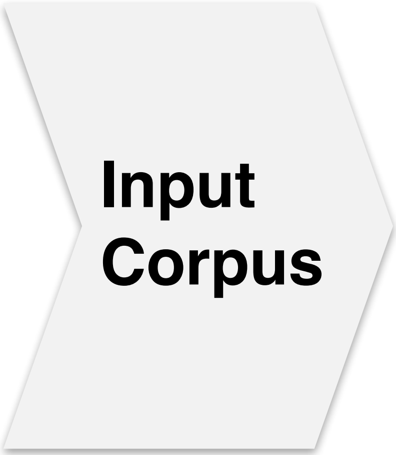
- Cleaning with RegExp
- Lemmatization
- Tokenization
- Lowercasing

English dictionaries

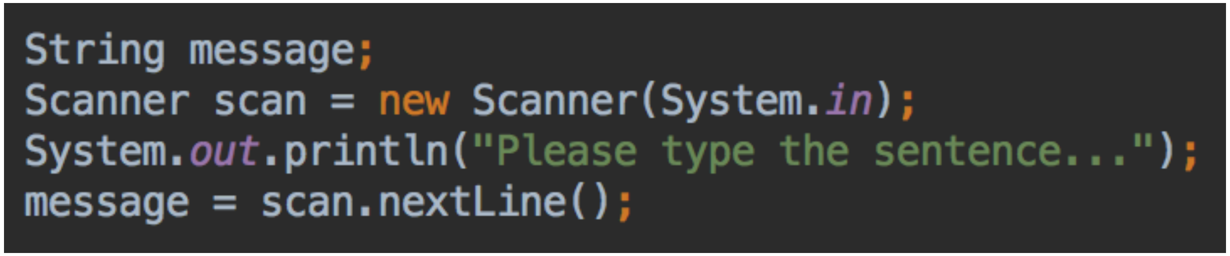
- THE DEVIL'S DICTIONARY ((C)1911 Released April 15 1993)
- Easton's 1897 Bible Dictionary
- Elements database 20001107
- The Free On-line Dictionary of Computing (27 SEP 03)
- U.S. Gazetteer (1990)
- The Collaborative International Dictionary of English v.0.44
- Hitchcock's Bible Names Dictionary (late 1800's)
- Jargon File (4.3.1, 29 June 2001)
- Virtual Entity of Relevant Acronyms (Version 1.9, June 2002)
- WordNet (r) 2.0
- CIA World Factbook 2002
- User Dictionary

Data processing



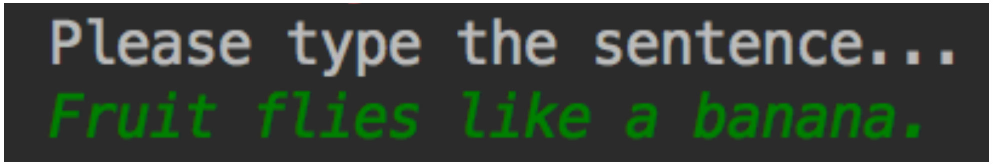


✓ **Java Code**



```
String message;
Scanner scan = new Scanner(System.in);
System.out.println("Please type the sentence...");
message = scan.nextLine();
```

✓ **Out Put**



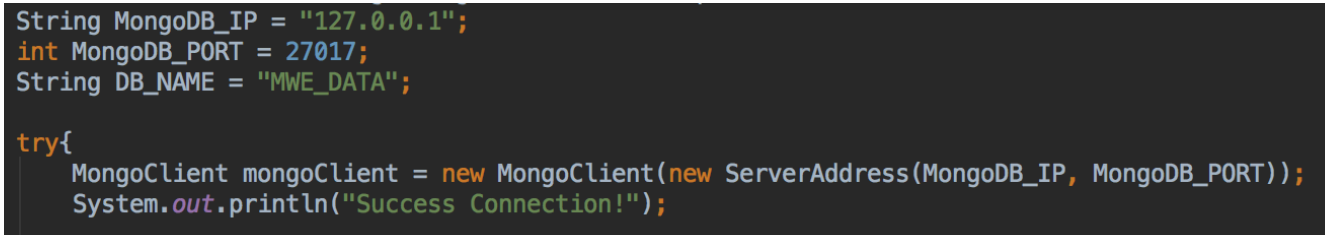
```
Please type the sentence...
Fruit flies like a banana.
```

Data processing

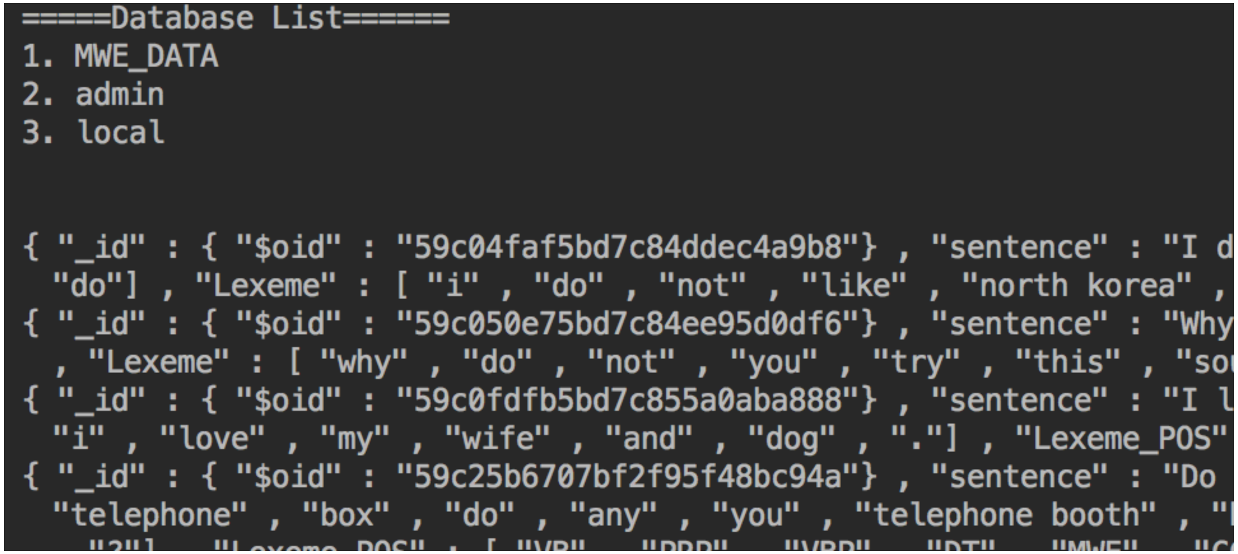
Input Corpus
Distinguish Sentence
Processing
Words candidate
Words validation
Storing Results

Distinguish Sentence

✓ MongoDB & JAVA



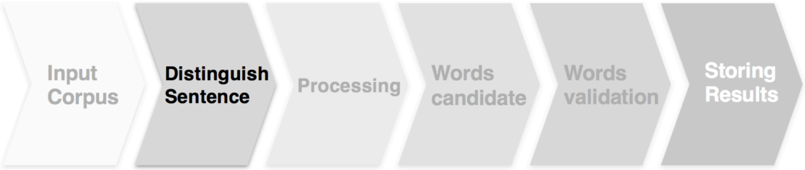
```
String MongoDB_IP = "127.0.0.1";
int MongoDB_PORT = 27017;
String DB_NAME = "MWE_DATA";

try{
    MongoClient mongoClient = new MongoClient(new ServerAddress(MongoDB_IP, MongoDB_PORT));
    System.out.println("Success Connection!");
```



```
=====Database List======
1. MWE_DATA
2. admin
3. local


{ "_id" : { "$oid" : "59c04faf5bd7c84ddec4a9b8"} , "sentence" : "I d
  "do"] , "Lexeme" : [ "i" , "do" , "not" , "like" , "north korea" ,
{ "_id" : { "$oid" : "59c050e75bd7c84ee95d0df6"} , "sentence" : "Why
  , "Lexeme" : [ "why" , "do" , "not" , "you" , "try" , "this" , "so
{ "_id" : { "$oid" : "59c0fdfb5bd7c855a0aba888"} , "sentence" : "I l
  "i" , "love" , "my" , "wife" , "and" , "dog" , "."] , "Lexeme_POS"
{ "_id" : { "$oid" : "59c25b6707bf2f95f48bc94a"} , "sentence" : "Do
  "telephone" , "box" , "do" , "any" , "you" , "telephone booth" , "
  "?"] , "Lexeme_POS" : [ "VB" , "PRP" , "VBP" , "DT" , "MWE" , "C
```

Université Paris Nanterre AJOU UNIVERSITY

Data processing



Input Corpus

Distinguish Sentence

Processing

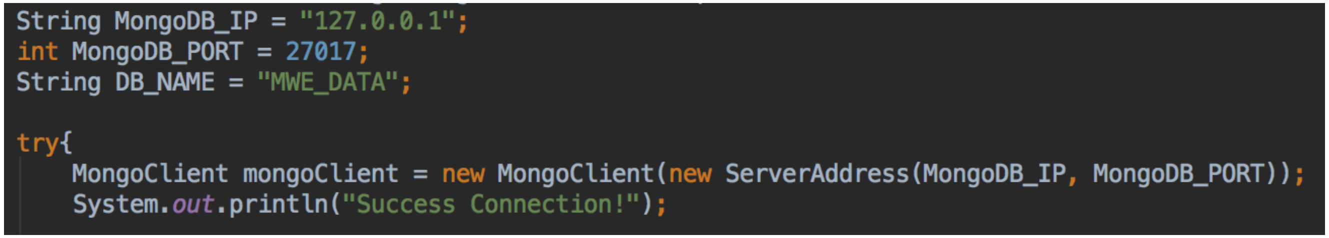
Words candidate

Words validation

Storing Results

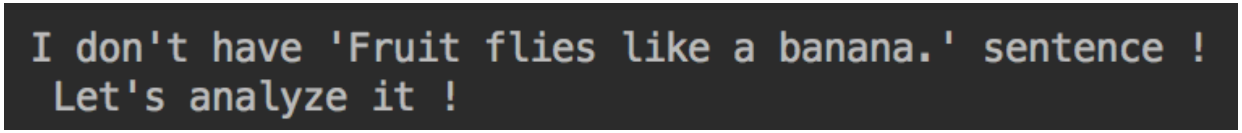
Distinguish Sentence

✓ **MongoDB & JAVA**



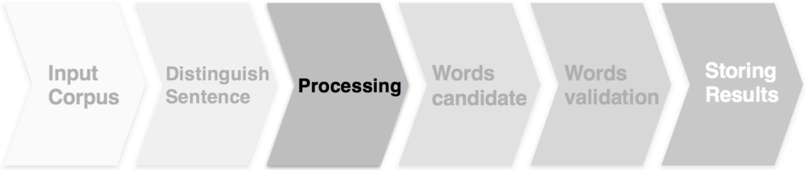
```
String MongoDB_IP = "127.0.0.1";
int MongoDB_PORT = 27017;
String DB_NAME = "MWE_DATA";

try{
    MongoClient mongoClient = new MongoClient(new ServerAddress(MongoDB_IP, MongoDB_PORT));
    System.out.println("Success Connection!");
```

✓ **Out Put**



```
I don't have 'Fruit flies like a banana.' sentence !
 Let's analyze it !
```

Data processing



Processing

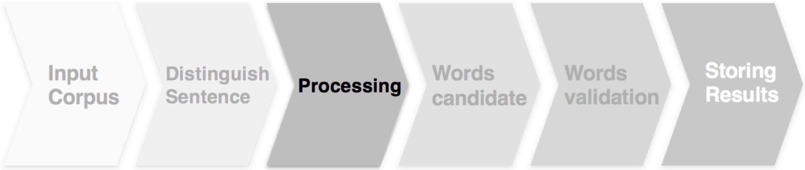
✓ **N-gram**

N-gram method is a contiguous sequence of ***N*** items from a given sequence of text.

✓ **Dependency Parser**

Dependency parser can provide a simple description of the grammatical relationships in a sentence.

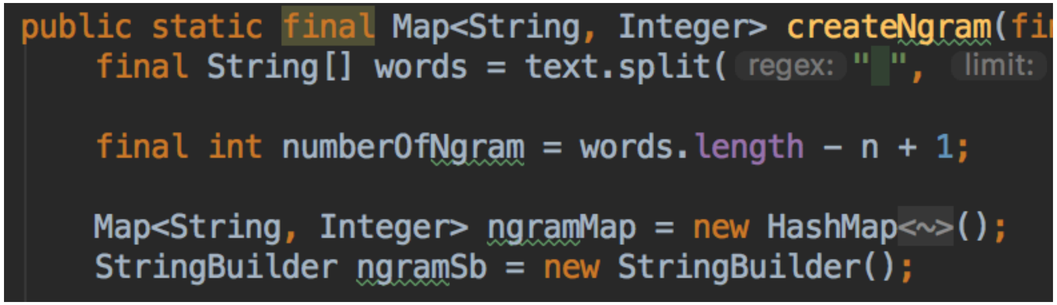
Data processing



Processing

✓ **N-gram**

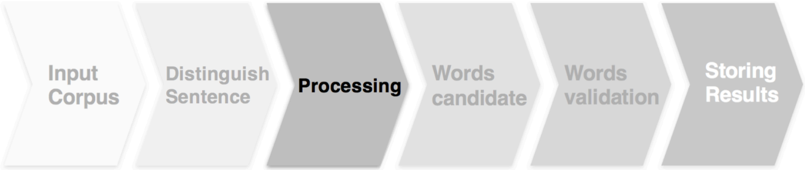
✓ **Java Code**



```
public static final Map<String, Integer> createNgram(fi
    final String[] words = text.split( regex: " ",  limit:

    final int numberOfNgram = words.length - n + 1;

    Map<String, Integer> ngramMap = new HashMap<~>();
    StringBuilder ngramSb = new StringBuilder();
```

Université Paris Nanterre AJOU UNIVERSITY

Data processing


Input Corpus
Distinguish Sentence
Processing
Words candidate
Words validation
Storing Results

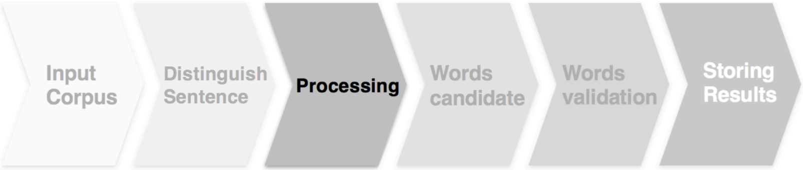
Processing

✓ **N-gram**

“Shall I wake him up?”

Unigram	:	Shall, I, wake, him, up.
Bigram	:	Shall I, I wake, wake him, him up.
Trigram	:	Shall I wake, I wake him, wake him up.

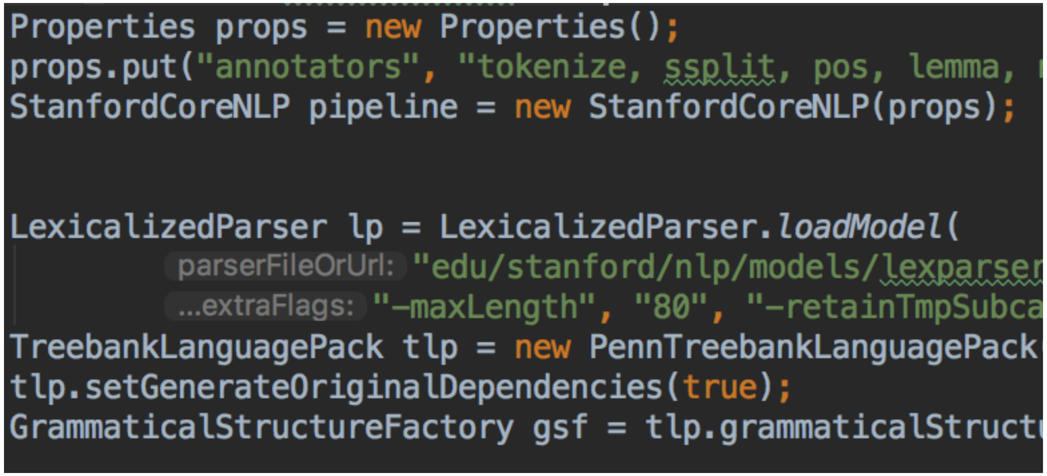
Data processing



Processing

✓ **Dependency parser**

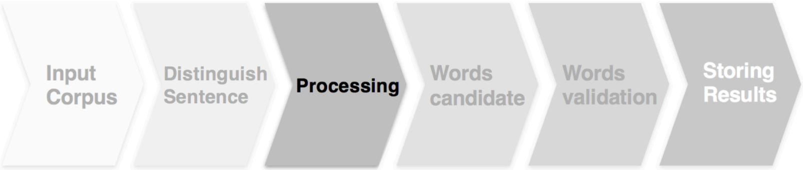
✓ **Java Code # Stanford_CoreNLP**



```
Properties props = new Properties();
props.put("annotators", "tokenize, ssplit, pos, lemma, 
StanfordCoreNLP pipeline = new StanfordCoreNLP(props);


LexicalizedParser lp = LexicalizedParser.loadModel(
        parserFileOrUrl: "edu/stanford/nlp/models/lexparser
        ...extraFlags: "-maxLength", "80", "-retainTmpSubca
TreebankLanguagePack tlp = new PennTreebankLanguagePack
tlp.setGenerateOriginalDependencies(true);
GrammaticalStructureFactory gsf = tlp.grammaticalStruct
```

Université Paris Nanterre AJOU UNIVERSITY

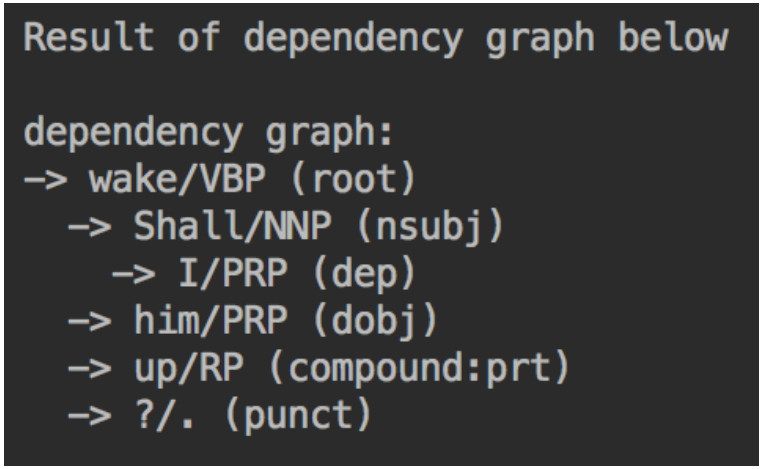
Data processing



Processing

✓ **Dependency parser**

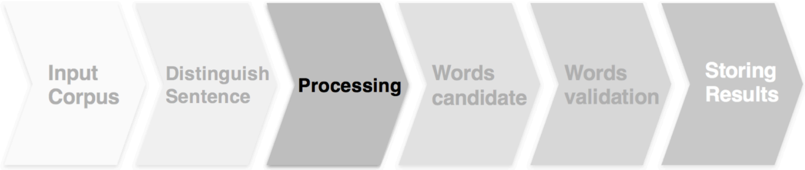
“Shall I wake him up?”



```
Result of dependency graph below

dependency graph:
-> wake/VBP (root)
  -> Shall/NNP (nsubj)
    -> I/PRP (dep)
  -> him/PRP (dobj)
  -> up/RP (compound:prt)
  -> ?/. (punct)
```

Université Paris Nanterre AJOU UNIVERSITY

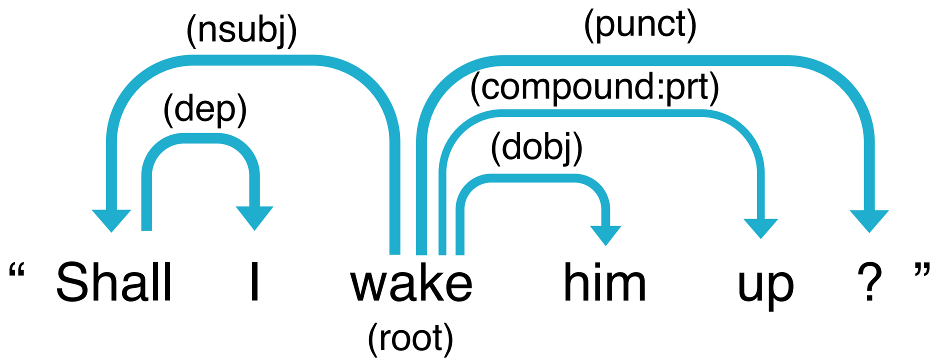
Data processing



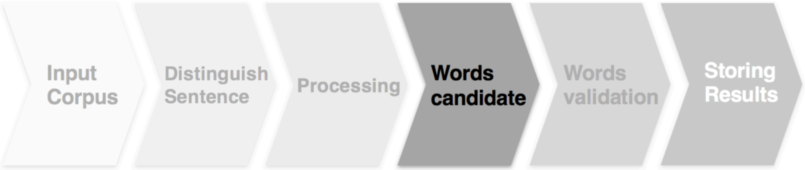
Processing

✓ **Dependency parser**

“Shall I wake him up?”

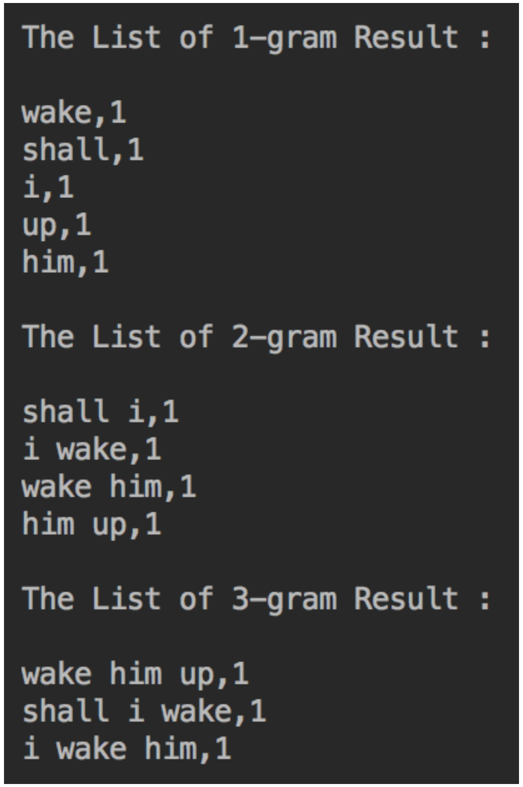


Data processing



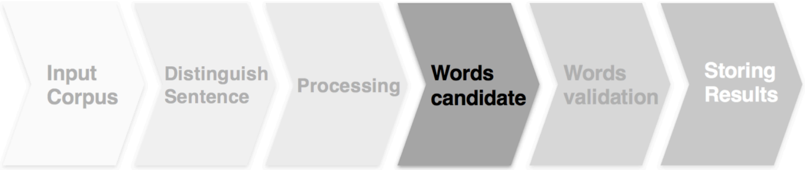
Words candidate

✓ **N-gram** Sentence : **“Shall I wake him up?”**



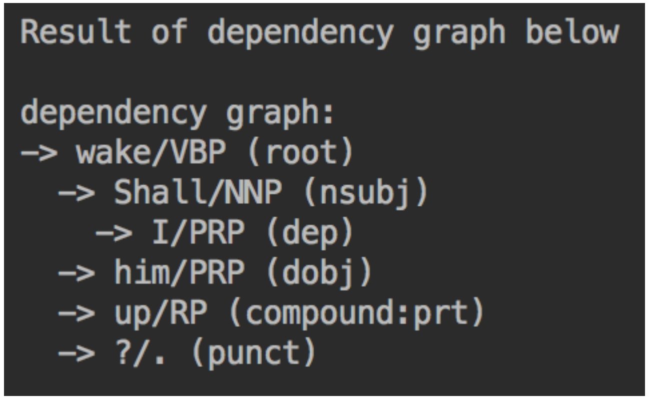
```
The List of 1-gram Result :

wake,1
shall,1
i,1
up,1
him,1

The List of 2-gram Result :

shall i,1
i wake,1
wake him,1
him up,1

The List of 3-gram Result :

wake him up,1
shall i wake,1
i wake him,1
```

Data processing

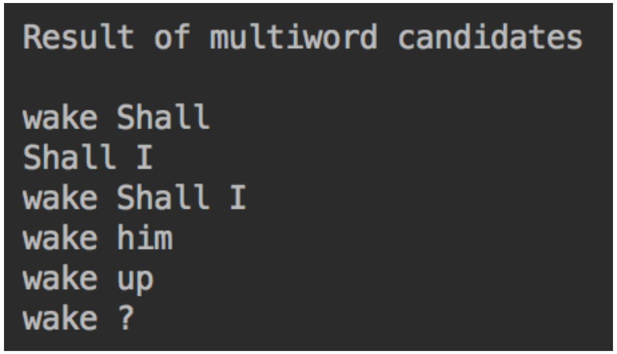


Words candidate

✓ **Dependency parser** Sentence : **“Shall I wake him up?”**

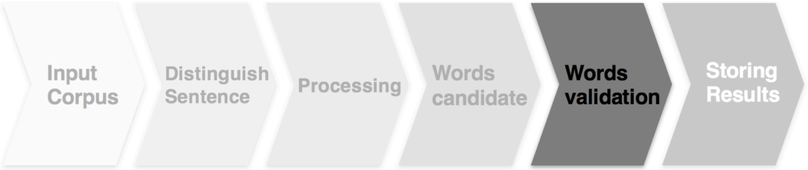


```
Result of dependency graph below

dependency graph:
-> wake/VBP (root)
  -> Shall/NNP (nsubj)
    -> I/PRP (dep)
  -> him/PRP (dobj)
  -> up/RP (compound:prt)
  -> ?/. (punct)
```



```
Result of multiword candidates

wake Shall
Shall I
wake Shall I
wake him
wake up
wake ?
```

Data processing



Words validation

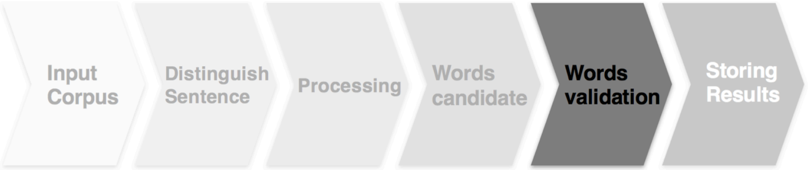
✓ **English Dictionaries**

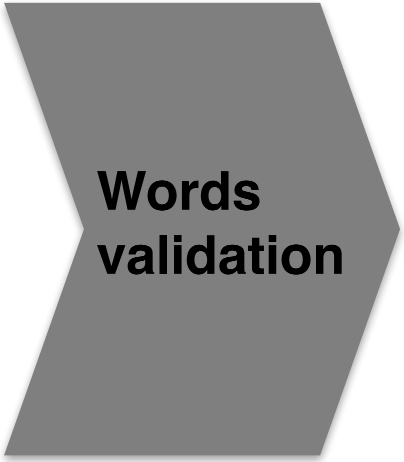
English dictionaries

- THE DEVIL'S DICTIONARY ((C)1911 Released April 15 1993)
- Easton's 1897 Bible Dictionary
- Elements database 20001107
- The Free On-line Dictionary of Computing (27 SEP 03)
- U.S. Gazetteer (1990)
- The Collaborative International Dictionary of English v.0.44
- Hitchcock's Bible Names Dictionary (late 1800's)
- Jargon File (4.3.1, 29 June 2001)
- Virtual Entity of Relevant Acronyms (Version 1.9, June 2002)
- WordNet (r) 2.0
- CIA World Factbook 2002
- User Dictionary

API : http://services.aonaware.com/DictService/

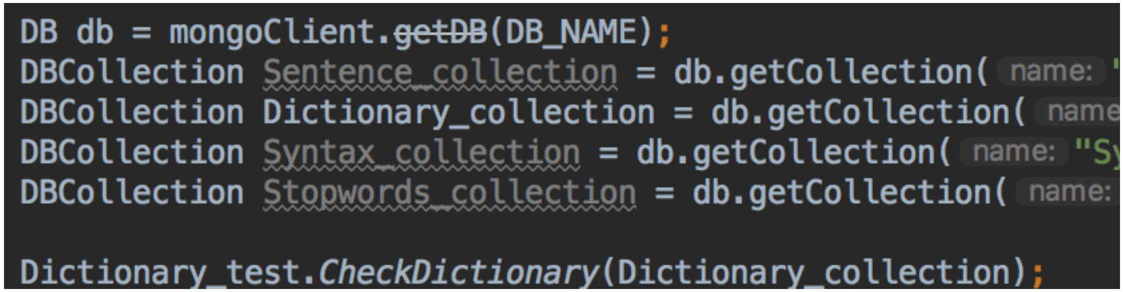
Data processing





✓ **User Dictionary**

✓ MongoDB & JAVA



```
DB db = mongoClient.~~getDB~~(DB_NAME);
DBCollection Sentence_collection = db.getCollection( name: '
DBCollection Dictionary_collection = db.getCollection( name
DBCollection Syntax_collection = db.getCollection( name: "Sy
DBCollection Stopwords_collection = db.getCollection( name:

Dictionary_test.CheckDictionary(Dictionary_collection);
```

✓ Detail :

{ "_id" : { "$oid" : ”Unique number"} , "word" : "" , "meaning" : ""}

Université Paris Nanterre AJOU UNIVERSITY

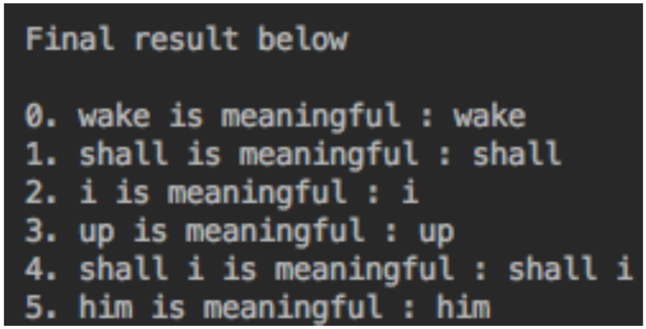
Data processing

Input Corpus
Distinguish Sentence
Processing
Words candidate
Words validation
Storing Results

Words validation

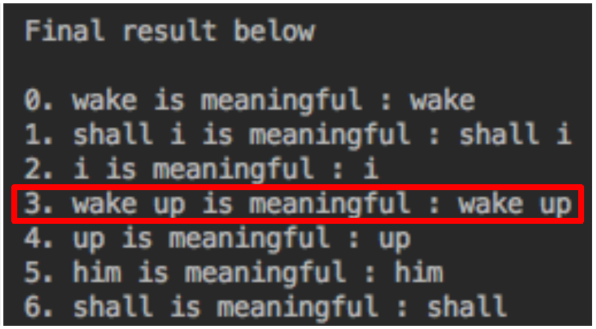
✓ **Accuracy** Sentence : “**Shall I wake him up?**”

✓ **N-gram & Dependency parser**



```
Final result below

0. wake is meaningful : wake
1. shall is meaningful : shall
2. i is meaningful : i
3. up is meaningful : up
4. shall i is meaningful : shall i
5. him is meaningful : him
```

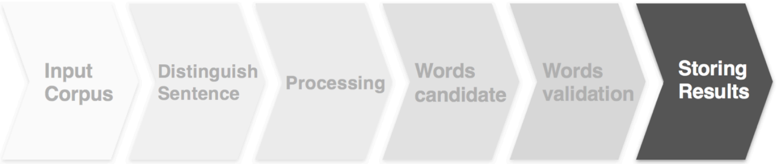
N-gram



```
Final result below

0. wake is meaningful : wake
1. shall i is meaningful : shall i
2. i is meaningful : i
3. wake up is meaningful : wake up
4. up is meaningful : up
5. him is meaningful : him
6. shall is meaningful : shall
```

Dependency graph + N-gram

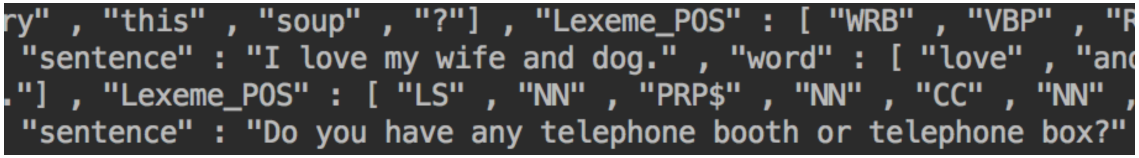
Data processing



Storing Results

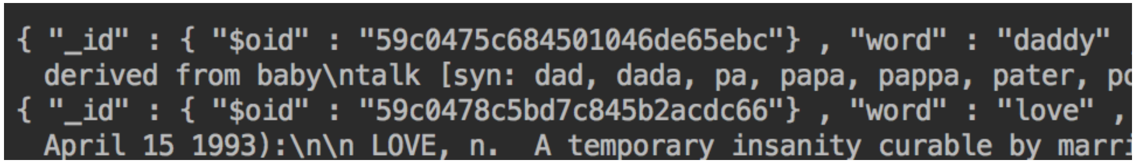
✓ Data Base : MongoDB & JAVA

✓ Sentence Collection



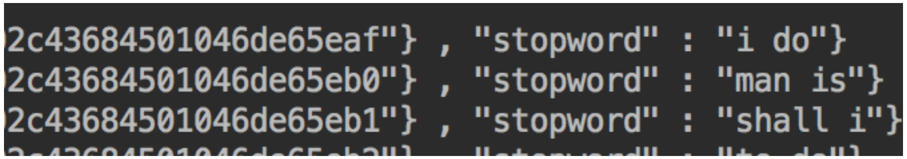
```
ry" , "this" , "soup" , "?"] , "Lexeme_POS" : [ "WRB" , "VBP" , "R
 "sentence" : "I love my wife and dog." , "word" : [ "love" , "and
."] , "Lexeme_POS" : [ "LS" , "NN" , "PRP$" , "NN" , "CC" , "NN" ,
 "sentence" : "Do you have any telephone booth or telephone box?"
```

✓ Dictionary Collection



```
{ "_id" : { "$oid" : "59c0475c684501046de65ebc"} , "word" : "daddy"
  derived from baby\ntalk [syn: dad, dada, pa, papa, pappa, pater, po
{ "_id" : { "$oid" : "59c0478c5bd7c845b2acdc66"} , "word" : "love" ,
  April 15 1993):\n\n LOVE, n.  A temporary insanity curable by marri
```

✓ Stopwords Collection



```
2c43684501046de65eaf"} , "stopword" : "i do"}
2c43684501046de65eb0"} , "stopword" : "man is"}
2c43684501046de65eb1"} , "stopword" : "shall i"}
2c43684501046de65eb2"} , "stopword" : "to do"}
```

Q&A

Thank you for listening.

stat34@ajou.ac.kr