How can we capture multiword expressions?

Seongmin Mun\textsuperscript{1}, Guillaume Desagulier\textsuperscript{2}, Kyungwon Lee\textsuperscript{3}

\textsuperscript{1} Lifemedia Interdisciplinary Program, Ajou University, South Korea
\textsuperscript{2} UMR 7114 MoDyCo - CNRS, University Paris Nanterre, France
\textsuperscript{3} Department of Digital Media, Ajou University, South Korea
Introduction

Words in a text corpus include features and information.

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

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
“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 (©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

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

✓ Output

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

✓ MongoDB & JAVA

```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" : { "$oid" : "59c04faf5bd7c84ddec4a9b8" }, "sentence" : "I do do", "Lexeme" : [ "i", "do", "not", "like", "north korea" ],
{ "_id" : { "$oid" : "59c050e75bd7c84ee95d0df6" }, "sentence" : "Why do you try", "Lexeme" : [ "why", "do", "not", "you", "try", "this", "so" ],
{ "_id" : { "$oid" : "59c0fdfb5bd7c855a0aba888" }, "sentence" : "I love my wife", "Lexeme_POS" : [ "i", "love", "my", "wife", "and", "dog", "" ],
{ "_id" : { "$oid" : "59c25b6707bf2f95f48bc94a" }, "sentence" : "Do you telephone", "Lexeme_POS" : [ "do", "telephone", "box", "do", "any", "you", "telephone booth", "" ] }
Data processing

✔ MongoDB & JAVA

```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!"神通);
}
```

✔ Out Put

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

- N-gram

Java Code

```java
public static final Map<String, Integer> createNgram(final String text, final int n) {
    final String[] words = text.split(regex: " ", limit: 2000);
    final int numberOfNgram = words.length - n + 1;

    Map<String, Integer> ngramMap = new HashMap<>();
    StringBuilder ngramSb = new StringBuilder();
    for (int i = 0; i < numberOfNgram; i++) {
        ngramMap.put(ngramSb.toString(), i);
        ngramSb.append(words[i]).append(' ');
    }
    return ngramMap;
}
```
Data 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

☑ Dependency parser

☑ Java Code # Stanford_CoreNLP

```java
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.grammaticalStructura
```
Data 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)
Data processing

✔ Dependency parser

“Shall I wake him up?”

“Shall I wake him up?”

(nsubj)

(dep)

(compound:prt)

(dobj)

(root)

(punct)

(dobj)
Data processing

✓ 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
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

✓ 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

```java
DB db = mongoClient.getDB(DB_NAME);
DBCollection Sentence_collection = db.getCollection( name: "Sentence" );
DBCollection Dictionary_collection = db.getCollection( name: "Dictionary" );
DBCollection Syntax_collection = db.getCollection( name: "Syntax" );
DBCollection Stopwords_collection = db.getCollection( name: "Stopwords" );

Dictionary_test.CheckDictionary(Dictionary_collection);
```

✓ Detail:

```json
{ "_id" : { "$oid" : "Unique number" }, "word" : "", "meaning" : "" }
```
Data processing

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

✓ **N-gram & Dependency parser**

<table>
<thead>
<tr>
<th>N-gram</th>
<th>Dependency graph + N-gram</th>
</tr>
</thead>
<tbody>
<tr>
<td>0. wake is meaningful : wake</td>
<td>0. wake is meaningful : wake</td>
</tr>
<tr>
<td>1. shall is meaningful : shall</td>
<td>1. shall i is meaningful : shall i</td>
</tr>
<tr>
<td>2. i is meaningful : i</td>
<td>2. i is meaningful : i</td>
</tr>
<tr>
<td>3. up is meaningful : up</td>
<td>3. wake up is meaningful : wake up</td>
</tr>
<tr>
<td>4. shall i is meaningful : shall</td>
<td>4. up is meaningful : up</td>
</tr>
<tr>
<td>5. him is meaningful : him</td>
<td>5. him is meaningful : him</td>
</tr>
<tr>
<td>6. shall is meaningful : shall</td>
<td>6. shall is meaningful : shall</td>
</tr>
</tbody>
</table>
Data processing

- Data Base: MongoDB & JAVA

- Sentence Collection
  ```json
  { "_id" : { "$oid" : "59c0475c684501046de65ebc" }, "word" : "daddy" derived from baby\ntalk [syn: dad, dada, pa, papa, pappa, pater, pop, }
  { "_id" : { "$oid" : "59c0478c5bd7c845b2accc66" }, "word" : "love", April 15 1993):\n  LOVE, n. A temporary insanity curable by marri
  ```

- Dictionary Collection
  ```javascript
  2c43684501046de65eaf" }, "stopword" : "i do"
  2c43684501046de65eb0" }, "stopword" : "man is"
  2c43684501046de65eb1" }, "stopword" : "shall i"
  ```

- Stopwords Collection
Thank you for listening.

stat34@ajou.ac.kr