Biomedical Text Mining and Natural Language Processing Workshop

UP-MIT-Stanford-AeHIN Big Data for Health Conference and Workshops for Asia-Pacific

http://ckbjimmy.github.io/2017_cebu

Wei-Hung Weng, MD, MMSc

July 5, 2017

CSAIL
Massachusetts Institute of Technology

Agenda

- We have 3.5 hours (13:30-17:00)
- 20-30 minutes introduction I
- 1 hour hands-on exercise
  - Regular expression
  - Language modeling 1: Bag-of-words / n-grams
- 20-30 minutes introduction II
- 1 hour hands-on exercise in R language
  - Language modeling 2: Topic modeling (LDA)
  - Language modeling 3: Word embedding (GloVe)
  - Language modeling 4: Hidden representation in the neural network (autoencoder)
- 20-30 minutes wrap-up

INTRODUCTION I

Dealing with Biomedical Text?

- Goal: Extracting previously unknown but important information (features)
- Data collection and preprocessing (maybe $>$ 80% of your time)
    - **Regular expression!**
- **Natural language processing**
- Exploratory analysis, statistics, missing value & outlier
- Annotation and analysis
- Modeling
- Evaluation
- Prediction

Difference between TM, IR and IE

- Typical **text mining** tasks include document classification, document clustering, building ontology, sentiment analysis, document summarization, information extraction, etc.
- **Information retrieval** typically deals with crawling, parsing and indexing document, retrieving documents
- **Information extraction** is the task of automatically extracting structured information from unstructured or semi-structured machine-readable documents.

Oliveira

Important Natural Language Features

- Part-Of-Speech Tagging (POS): syntactic roles (noun, adverb...)
- Chunking (CHUNK): syntactic constituents (noun phrase, verb phrase...)
- Name Entity Recognition (NER): person/company/location...
- Semantic Role Labeling (SRL): semantic role
- Word sense disambiguation (WSD)
- Co-reference resolution (pronoun)

Collobert, Weston 2009

More Features

Predicate and POS tag of predicate	Voice: active or passive (hand-built rules)
Phrase type: adverbial phrase, prepositional phrase, . . .	Governing category: Parent node's phrase type(s)
Head word and POS tag of the head word	Position: left or right of verb
Path: traversal from predicate to constituent	Predicted named entity class
Word-sense disambiguation of the verb	Verb clustering
Length of the target constituent (number of words)	NEG feature: whether the verb chunk has a "not"
Partial Path: lowest common ancestor in path	Head word replacement in prepositional phrases
First and last words and POS in constituents	Ordinal position from predicate + constituent type
Constituent tree distance	Temporal cue words (hand-built rules)
Dynamic class context: previous node labels	Constituent relative features: phrase type
Constituent relative features: head word	Constituent relative features: head word POS
Constituent relative features: siblings	Number of pirates existing in the world. . .

Collobert, Weston 2009

Pattern Matching

- Regular expression
  - A regular expression, regex or regexp (sometimes called a rational expression) is, in theoretical computer science and formal language theory, a sequence of characters that define a search pattern. Usually this pattern is then used by string searching algorithms for ”find” or ”find and replace” operations on strings.

Wikipedia

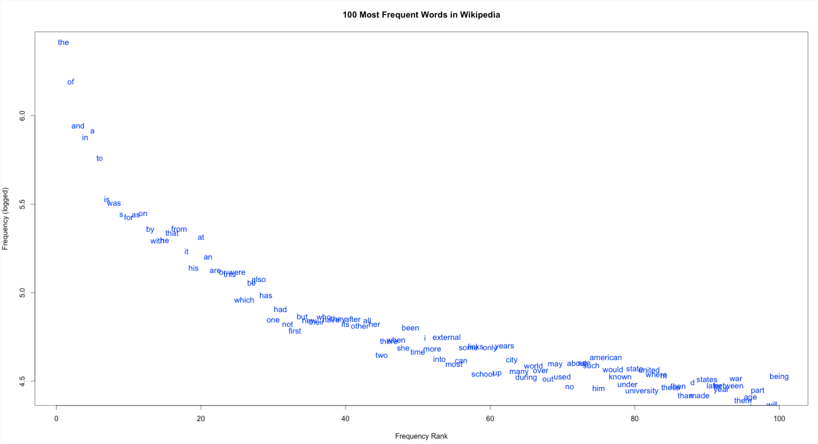
Text Processing

- Text segmentation / Tokenization
  - Alphabetic or Non-alphabetic (Chinese / Japanese / Tibetan...)
  - Separated characters may be meaningless
  - New York-New Haven (the same characters in different order)
- Stemming and Lemmatization (grammar)
  - Different words, same or similar meanings
  - 'imaging', 'imagination', 'image'
  - 'be', 'am', 'is', 'are'
- Part-of-speech (POS) tagging
  - NN, VV, ...
  - For semantic analysis
- Removing stopwords
  - Frequent but meaningless or not important

Natural Language Modeling

- Bag-of-words
  - One-hot encoding representation
  - Simple but useful
  - Frequency $\propto$ representative?
  - Zipf's Law (Zipf 1949)
  - Words with high term frequencies may be just common terms
  - Tf-idf: importance estimation
  - Problem: no word sequence meaning

Zipf’s Law



http://wugology.com/zipfs-law/

Natural Language Modeling

- n-gram model
  - Google Ngram Viewer
  - Consecutive n words
  - Some words are meaningful only when they are observed together
  - Information of phrase
  - Bag-of-words (unigram) $\rightarrow$ n-grams
    - I like dog
    - BoW: ['I', 'like', 'dog']
    - BoW + n-gram: ['I', 'like', 'dog', 'I like', 'like dog', 'I like dog'] (unigram + bigram + trigram)

Tf-idf Weighting

- Importance of the term in the corpus
- For term i in document j

$$w_{i,j} = tf_{i,j} \times \log(\frac{N}{df_i})$$

- $tf_{i,j}$: frequency of i in j
- df_i: number of documents have i
- N: number of all documents

Tf-idf Weighting Example

- A: ”Dog is so cute.”
- B: ”I like dog.”
- $tfidf_{('dog',A)} = \frac{1}{4} \times \log(\frac{2}{2}) = 0$
- $tfidf_{('dog',B)} = \frac{1}{3} \times \log(\frac{2}{2}) = 0$
- $tfidf_{('cute',A)} = \frac{1}{4} \times \log(\frac{2}{1}) = \frac{\log 2}{4}$
- $tfidf_{('cute',B)} = \frac{0}{3} \times \log(\frac{2}{0}) = 0$

HANDS-ON EXERCISE I

Link

http://ckbjimmy.github.io/2017_cebu

Problem / Scenario 1

- Want to replace/extract strings with specific patterns from free text
  - e.g. Extract ”chief complaints” from admission notes
  - e.g. Extract ”diagnosis” from pathology reports
  - e.g. Extract SBP/DBP as features for machine learning
- Solution
  - Regular expression

Regular Expression

- Crazy regex
- Some tools that can help you
    - regex101
    - regexr
- Regex cheatsheet
- Also a cheatsheet

E.g. ’echocardiogram’

Pattern	Meaning	Example
.	all characters	echocardiogram
cardi	phrase ’cardi’	cardi
.*cardi	0 or more characters before	echocardi
[a-z]*cardi	0+ lower case (only) before	echocardi
[A-Z]*cardi	0+ upper case (only) before	cardi
[aeiou]*cardi	0+ aeiou (only) before	ocardi
[aA-zZ]+cardi	if we use ’xcardiogram’	xcardi
[aA-zZ]{2,}cardi	if we use ’xcardiogram’	-
cardi|gram	catches ’cardi’ or ’gram’	cardi, gram
\d	catches any digit	-
\d3, 5	catches 3 to 5 digits	-

Problem / Scenario 2

- Want to build a machine learning model using all information from the whole dataset
- Want to know the characteristics of the dataset
  - e.g. Understand the word appearance/frequency among documents
  - e.g. Cluster/Classify the documents
  - e.g. Find the semantics of words inside the corpus
- Solution
  - Build language models (bag-of-words, n-grams, ...), perform data exploration and visualization, and use the model for machine learning

Building Simple Language Model Using R

- `tm` package in R (Feinerer, Hornik 2014)
- Steps
    1. Convert to lower case
    2. Remove punctuation, numbers, URLs, emoji
    3. Remove stopwords
    4. Lemmatization, stemming
    5. Tokenization
    6. POS tagging (optional, need to use `openNLP` package)
    7. Tf-idf weighting
    8. n-grams
    9. Convert to document-term matrix

Exploration and Visualization in R

- Wordcloud (`wordcloud`)
- Frequency plot (`ggplot2`)
- Unsupervised clustering
    - k-means clustering (`fpc, cluster`)

INTRODUCTION II

Learning Features?

- Large scale hand-made feature engineering!
- Task-specific engineering limits NLP scope
- We want to avoid task-specific engineering
- Can we find unified hidden representations? Can we build unified NLP architecture? Can we utilize/preserve semantics?
- More algorithmic approaches for knowledge representation!

Deciphering Hieroglyphs

(knife)	51	20	84	0	3	0
(cat)	52	58	4	4	6	26
???	115	83	10	42	33	17
(boat)	59	39	23	4	0	0
(cup)	98	14	6	2	1	0
(pig)	12	17	3	2	9	27
(banana)	11	2	2	0	18	0

Evert 2010

(knife)	51	20	84	0	3	0
(cat)	52	58	4	4	6	26
???	115	83	10	42	33	17
(boat)	59	39	23	4	0	0
(cup)	98	14	6	2	1	0
(pig)	12	17	3	2	9	27
(banana)	11	2	2	0	18	0

sim(,) = 0.770

Evert 2010

Deciphering Hieroglyphs

(knife)		51	20	84	0	3	0
(cat)		52	58	4	4	6	26
???		115	83	10	42	33	17
(boat)		59	39	23	4	0	0
(cup)		98	14	6	2	1	0
(pig)		12	17	3	2	9	27
(banana)		11	2	2	0	18	0

$$\mathrm{sim}(\;,\;) = 0.939$$

Evert 2010

Deciphering Hieroglyphs

(knife)	51	20	84	0	3	0
(cat)	52	58	4	4	6	26
???	115	83	10	42	33	17
(boat)	59	39	23	4	0	0
(cup)	98	14	6	2	1	0
(pig)	12	17	3	2	9	27
(banana)	11	2	2	0	18	0

sim(,) = 0.961

Evert 2010

Deciphering Hieroglyphs

	get	see	use	hear	eat	kill
knife	51	20	84	0	3	0
cat	52	58	4	4	6	26
dog	115	83	10	42	33	17
boat	59	39	23	4	0	0
cup	98	14	6	2	1	0
pig	12	17	3	2	9	27
banana	11	2	2	0	18	0

verb-object counts from British National Corpus

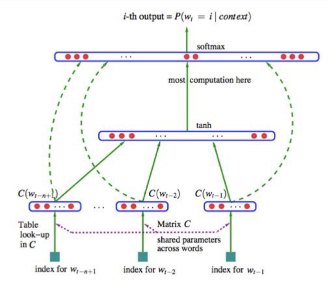
Evert 2010

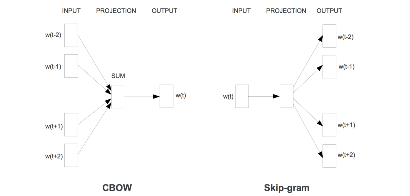
Semantic Approaches

- Matrix decomposition
    - LSI (Deerwester 1990), NMF (Lee 1999), NTF (Cruys 2010)
    - Using SVD ($U\Sigma V$)
    - Fast, unless using NTF
- Probabilistic language model
    - PLSI (Hofmann 1999), LDA (Blei 2003)
    - Topic modeling, using probability
    - Heavy computation

Neural Language Model

- NNLM (Bengio 2003), RNN/LSTM, autoencoder
- skip-gram / CBOW (word2vec, Mikolov 2013) for word embedding
- Heavy computation, hard to implementation
- Interpretation...?

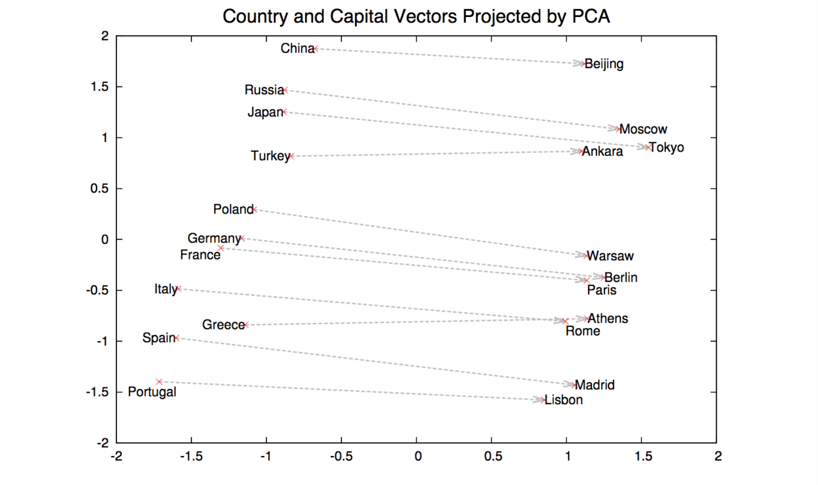




Bengio 2003

Mikolov 2013

An Example of Word Embedding



Mikolov 2013

HANDS-ON EXERCISE II

Link

http://ckbjimmy.github.io/2017_cebu

Problem / Scenario 3

- Want to categorize / organize documents in the dataset into ideal groups based on the theme of document
    - e.g. Organize clinical notes based on medical specialty
- Solution
    - Topic modeling

Problem / Scenario 4

- Want to know the ”word similarity”
- Want to utilize the word semantics for machine learning (e.g. convolutional neural network)
    - e.g. Which is likely, diabetes & nephrologist or diabetes & endocrinologist?
- Solution
    - Word embedding

Problem / Scenario 5

- Want to find a representation for all documents in the dataset
    - e.g. What's the similarity between this note and that report?
- Solution
    - Autoencoder / Deep autoencoder

Exercise

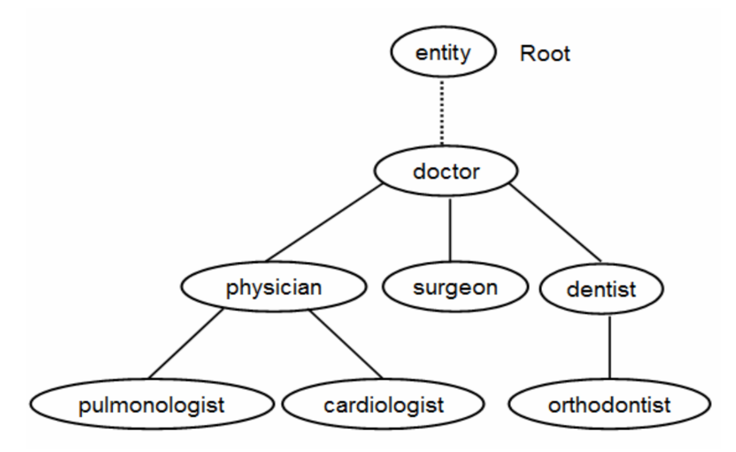
- Topic modeling using latent Dirichlet allocation (`topicmodels`)
- Word embedding using GloVe (`text2vec`)
- Extracting the hidden representation in deep autoencoder (`keras`)

UTILIZING HUMAN-CURATED KNOWLEDGE

Problem / Scenario 6

- All above methods are not reliable! No real expert knowledge in the model. I want to utilize expert knowledge instead of pure, arcane machine representation.
- Solution
    - Utilize medical ontology (e.g. RxNorm for medication, LOINC for procedure, ICD-10CM for diagnosis), and use cTAKES or MetaMap to identify ”real clinical terms” in the clinical documents

Ontology for Knowledge Representation



Liu, 2012

Medical Ontology

- SNOMED-CT (for all medical terms)
- RxNorm (for medication)
- MeSH (for all biomedical terms)
- ICD-10 (for disease categorization)
    - `W22.02XD: Walked into lamppost, subsequent encounter.`
    - `W59.29XS: Other contact with turtle, sequel.`
    - `V97.33XD: Sucked into jet engine, subsequent encounter.`
- FMA (for anatomy)
- HPO (for rare diseases)

Interconnectivity

- Upper level connection
- UMLS Metathesaurus
- Make sure you already have UTS account
- Two versions per year (now 2016AA)
- Concept unique identifier (CUI)
  - `C0031511|SNOMEDCT_US|154555009|Phaeochr...`
  - `C0031511|SCTSPA|85583005|feocromocitoma`

Semantic Connection

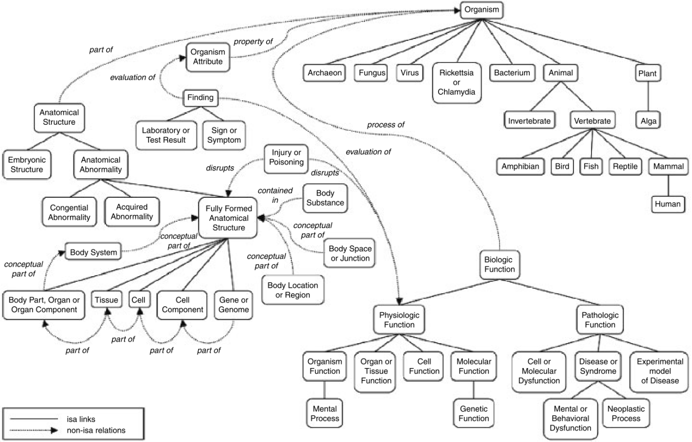
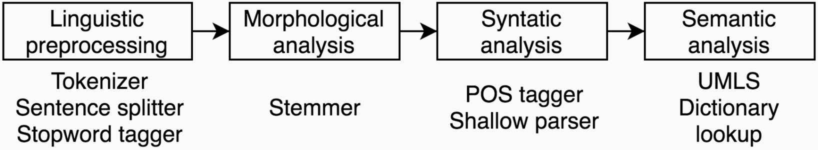


Figure 1. A portion of the UMLS semantic network

McCray, 2003

Some Resources

- BioPortal ontology repository
- UMLS
  - UTS web application for UMLS
- SNOMED
  - UTS web application for SNOMED
- RxNorm
  - RxNav (Web application for RxNorm)
- LOINC
- Human Phenotype Ontology
  - For rare, congenital diseases



MetaMap / cTAKES workflow

MetaMap

- Developed by NLM (Aronson 1994)
- Web application of MetaMap
- Java API
- Locally execution
- Download

MetaMap

```
Control options:
  composite_phrases=4
  lexicon=db
  mm_data_year=2015AB
|: thoracentesis was done 1000cc of amber fluid obtained sent spec to lab.pt has another very large formed stool.tube feeds were restarted due
to trach to be done in am not today.peg also planed for future.family spoke with md
|:
Processing 00000000.tx.1: thoracentesis was done 1000cc of amber fluid obtained sent spec to lab.pt has another very large formed stool.tube fe
eds were restarted due to trach to be done in am not today.peg also planed for future.family spoke with md

Phrase: thoracentesis
Meta Mapping (1000):
  1000   Thoracentesis [Diagnostic Procedure,Therapeutic or Preventive Procedure]

Phrase: was

Phrase: done

Phrase: 1000cc of amber fluid
Meta Mapping (555):
   604   AMBER (Amber) [Organic Chemical]
   604   FLUID (Body Fluids) [Body Substance]
```

```
weng-2:bin weng$ echo thoracentesis was done 1000cc of amber fluid obtained sent spec to lab.pt has another very large formed stool.tube feeds
were restarted due to trach to be done in am not today.peg also planed for future.family spoke with md | ./metamap -N --prune 20
```

```
  prune=20
00000000|MMI|20.95|Thoracentesis|C0189477|[diap,topp]|["Thoracentesis"-tx-1-"thoracentesis"-noun-0]|TX|0/13|E01.370.225.998.329.810;E02.800.550
.810;E04.665.600.810;E05.200.998.329.810
00000000|MMI|17.80|Togo|C0040363|[geoa]|["TO"-tx-1-"to"-adv-0]|TX|149/2|Z01.058.290.190.800
00000000|MMI|17.80|Tryptophanase|C0041260|[aapp,enzy]|["TO"-tx-1-"to"-adv-0]|TX|149/2|D08.811.520.224.800
00000000|MMI|16.05|Amber|C0242864|[orch]|["AMBER"-tx-1-"amber"-adj-0]|TX|33/5|D05.750.078.840.109;D20.215.721.500.109
00000000|MMI|13.02|Family|C0015576|[famg]|["Family"-tx-1-"family"-noun-0]|TX|203/6|F01.829.263;I01.880.853.150
00000000|MMI|9.74|Laboratory|C0022877|[mnob,orgt]|["Lab"-tx-1-"lab"-noun-0]|TX|67/3|N02.278.487
00000000|MMI|6.77|Feces|C0015733|[bdsu]|["STOOL"-tx-1-"stool"-noun-0]|TX|104/5|A12.459
00000000|MMI|6.71|Future|C0016884|[tmco]|["Future"-tx-1-"future"-noun-0]|TX|196/6|I01.320
00000000|MMI|6.59|Body Fluids|C0005889|[bdsu]|["FLUID"-tx-1-"fluid"-noun-0]|TX|39/5|A12.207
00000000|MMI|5.18|Obtain|C1301820|[ftcn]|["Obtained"-tx-1-"obtained"-verb-0]|TX|45/8|
```

cTAKES

- Developed by Mayo NLP (Savova 2010)
- Modularized
- CLI
- Download

cTAKES


```xml
<org.apache.ctakes.typesystem.type.syntax.TreebankNode _indexed="1" _id="23431" _ref_sofa="3" begin="920" end="1078" nodeType="S" leaf="false"
_ref_parent="23413" _ref_children="23444" _ref_nodeTags="23442" headIndex="1"/>
<org.apache.ctakes.typesystem.type.syntax.TreebankNode _indexed="1" _id="23449" _ref_sofa="3" begin="920" end="925" nodeType="NP" leaf="false"
_ref_parent="23431" _ref_children="23463" _ref_nodeTags="23460" headIndex="0"/>
<org.apache.ctakes.typesystem.type.syntax.TreebankNode _indexed="1" _id="23466" _ref_sofa="3" begin="926" end="1077" nodeType="VP" leaf="false"
_ref_parent="23431" _ref_children="23479" _ref_nodeTags="23477" headIndex="1"/>
<org.apache.ctakes.typesystem.type.syntax.TreebankNode _indexed="1" _id="23483" _ref_sofa="3" begin="929" end="1077" nodeType="NP" leaf="false"
_ref_parent="23466" _ref_children="23497" _ref_nodeTags="23494" headIndex="5"/>
<org.apache.ctakes.typesystem.type.syntax.TreebankNode _indexed="1" _id="23501" _ref_sofa="3" begin="929" end="957" nodeType="NP" leaf="false"
_ref_parent="23483" _ref_children="23514" _ref_nodeTags="23512" headIndex="4"/>
<org.apache.ctakes.typesystem.type.syntax.TreebankNode _indexed="1" _id="23519" _ref_sofa="3" begin="958" end="1077" nodeType="VP" leaf="false"
_ref_parent="23483" _ref_children="23532" _ref_nodeTags="23530" headIndex="5"/>
<org.apache.ctakes.typesystem.type.syntax.TreebankNode _indexed="1" _id="23536" _ref_sofa="3" begin="968" end="1077" nodeType="PP" leaf="false"
_ref_parent="23519" _ref_children="23549" _ref_nodeTags="23547" headIndex="6"/>
<org.apache.ctakes.typesystem.type.syntax.TreebankNode _indexed="1" _id="23553" _ref_sofa="3" begin="971" end="1077" nodeType="NP" leaf="false"
_ref_parent="23536" _ref_children="23566" _ref_nodeTags="23564" headIndex="7"/>
```

```xml
<org.apache.ctakes.typesystem.type.refsem.UmlsConcept _id="6235" codingScheme="SNOMED" code="261405004" oid="261405004#SNOMED" score="0.0"
disambiguated="false" cui="C1280619" tui="T023"/>
<org.apache.ctakes.typesystem.type.refsem.UmlsConcept _id="6225" codingScheme="SNOMED" code="59652004" oid="59652004#SNOMED" score="0.0"
disambiguated="false" cui="C0018792" tui="T023"/>
<uima.cas.FSArray _id="6245" size="2">
    <i>6225</i>
    <i>6235</i>
</uima.cas.FSArray>
<org.apache.ctakes.typesystem.type.refsem.UmlsConcept _id="6189" codingScheme="SNOMED" code="244383003" oid="244383003#SNOMED" score="0.0"
disambiguated="false" cui="C1269890" tui="T023"/>
<org.apache.ctakes.typesystem.type.refsem.UmlsConcept _id="6169" codingScheme="SNOMED" code="264181001" oid="264181001#SNOMED" score="0.0"
disambiguated="false" cui="C0225844" tui="T023"/>
<org.apache.ctakes.typesystem.type.refsem.UmlsConcept _id="6179" codingScheme="SNOMED" code="73829009" oid="73829009#SNOMED" score="0.0"
disambiguated="false" cui="C0225844" tui="T023"/>
<uima.cas.FSArray _id="6199" size="3">
    <i>6179</i>
```

Some Advanced NLP Books / Online Courses

- NLTK book (useful for text preprocessing and traditional NLP)
- Foundations of Statistical Natural Language Processing (Manning)
- Speech and Language Processing (Jurafsky)
- Coursera NLP (Jurafusky)
- Coursera NLP (Radev)
- Coursera NLP provided by Michael Collins is also good, but it's gone now
- Stanford Natural Language Processing with Deep Learning
- Oxford Deep NLP

Take Home Message

- Text preprocessing and NLP for feature extraction
- Topic modeling
- Different language modeling techniques
  - Bag-of-words, n-grams, word embedding, autoencoder
- Unifying biomedical language
- Clinical NLP systems: MetaMap / cTAKES
- Contact
  - ckbjimmy@mit.edu
  - Linkedin: Wei-Hung Weng