Introduction to Text Mining

Yogesh Kulkarni

March 29, 2021

Information Extraction

"The task of Information Extraction (IE) involves extracting meaningful information from unstructured text data and presenting it in a structured format."

Example

We can extract the following information from the text:

> Indian captain Virat Kohli was dismissed cheaply for just 2 in Wellington on Friday by debutant Kyle Jamieson extending a rare lull in the batsman's stellar career. Throughout the ongoing New Zealand tour, Kohli has managed to score just a single fifty across 8 innings in all 3 international formats.

- Country – India, Captain – Virat Kohli
- Batsman – Virat Kohli, Runs – 2
- Bowler – Kyle Jamieson
- Match venue – Wellington
- Match series – New Zealand
- Series highlight – single fifty, 8 innings, 3 formats

Makes text structured meaning with relationships (key value).

(Ref: Hands-on NLP Project: A Comprehensive Guide to Information Extraction using Python - Aniruddha Bhandari)

Information Extraction Use case

As a Task

As a task: **Filling slots in a database from unstructured text.**

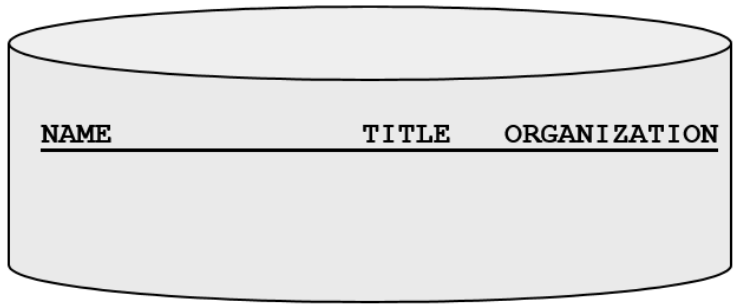
October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is [..].

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying...



Task

Task:

Extract structured data, such as tables, from unstructured text

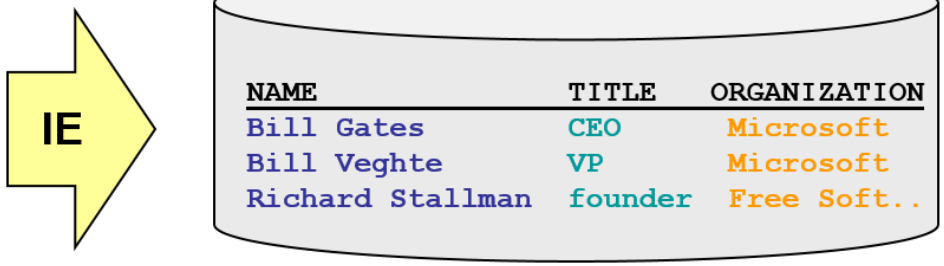
October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is [..].

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying...



Extraction

Information Extraction =
segmentation + classification + association

October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying…

Microsoft Corporation
CEO
Bill Gates
Microsoft
Gates
Microsoft
Bill Veghte
Microsoft
VP
Richard Stallman
founder
Free Software Foundation

aka "named entity extraction/detection"

Adapted from slide by William Cohen

Techniques

A family of techniques:

Information Extraction = segmentation + classification + association

October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying...

Microsoft Corporation
CEO
Bill Gates
Microsoft
Gates
Microsoft
Bill Veghte
Microsoft
VP
Richard Stallman
founder
Free Software Foundation

Example

Information Extraction =
segmentation + classification + association

October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying...

Microsoft Corporation
CEO
Bill Gates

Microsoft
Gates

Microsoft

Bill Veghte
Microsoft
VP

Richard Stallman
founder
Free Software Foundation

Adapted from slide by William Cohen

Landscape of Information Extraction

Web site specific

Formatting

Amazon.com Book Pages

Genre specific

Layout

Resumes

Wide, non-specific

Language

University Names

Adapted from slide by William Cohen

Complexity of Information Extraction

Closed set

U.S. states

He was born in Alabama…

The big Wyoming sky…

Regular set

U.S. phone numbers

Phone: (413) 545-1323

The CALD main office can be reached at 412-268-1299

Complex pattern

U.S. postal addresses

University of Arkansas
P.O. Box 140
Hope, AR 71802

Headquarters:
1128 Main Street, 4th Floor
Cincinnati, Ohio 45210

Ambiguous patterns, needing context and many sources of evidence

Person names

…was among the six houses sold by Hope Feldman that year.

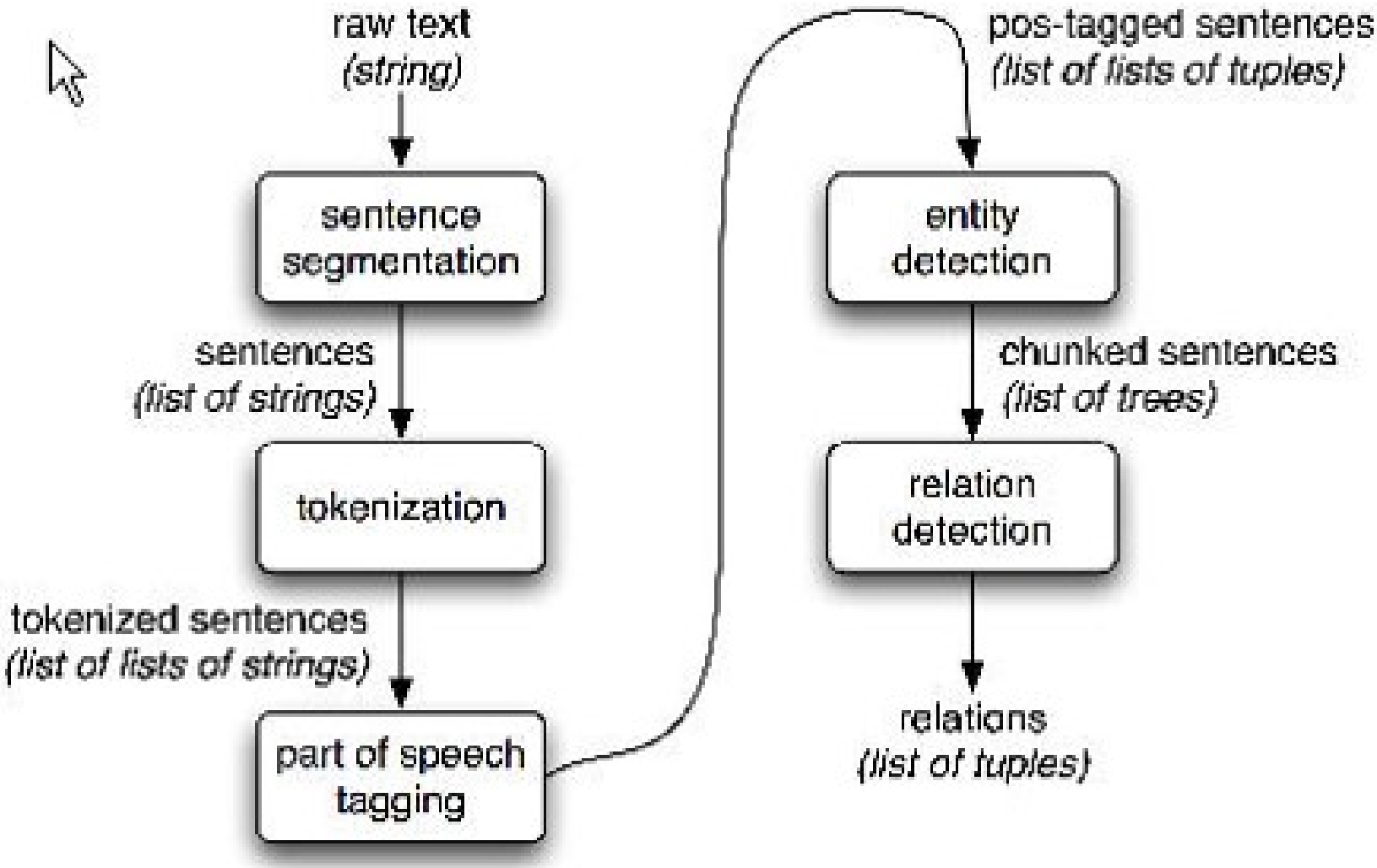
Pawel Opalinski, Software Engineer at WhizBang Labs.

Adapted from slide by William Cohen

Applications of Information Extraction

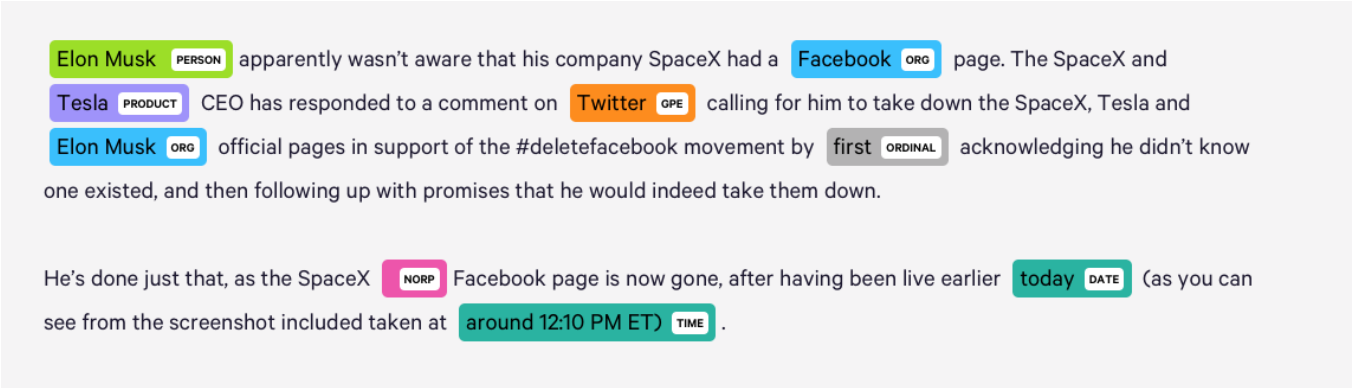
- Extract structured data out of electronically-available scientific literature, especially in the domain of biology and medicine
- Legal documents
- Business intelligence
- Resume harvesting
- Media analysis
- Sentiment detection
- Patent search
- Email scanning

Architecture of Information Extraction



Main tasks of Information Extraction

- Named Entity Recognition (Chunking, Parsing)
- Relation Extraction
- Relations like subject are syntactic, relations like person, location, agent or message are semantic

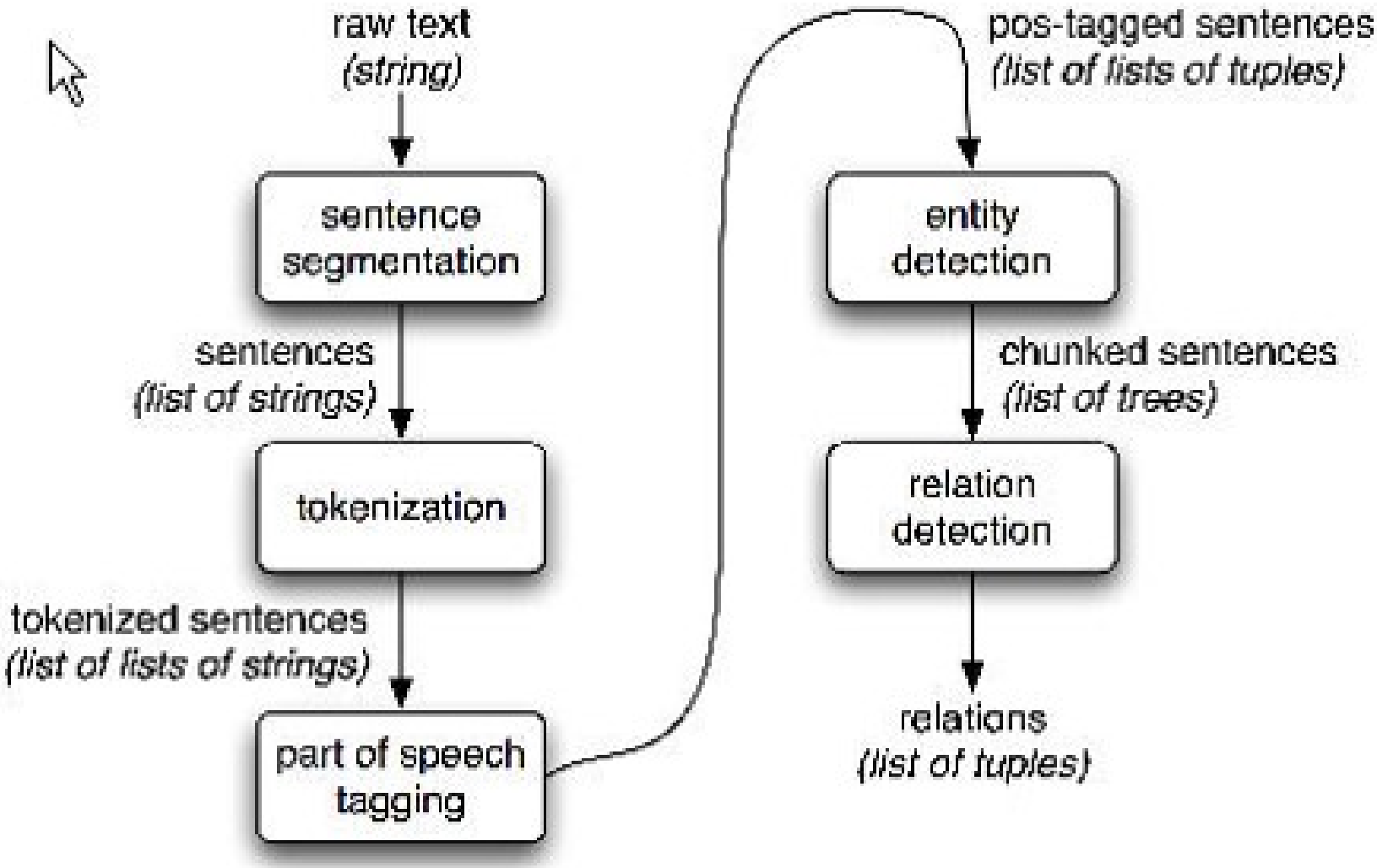


Example of Information Extraction

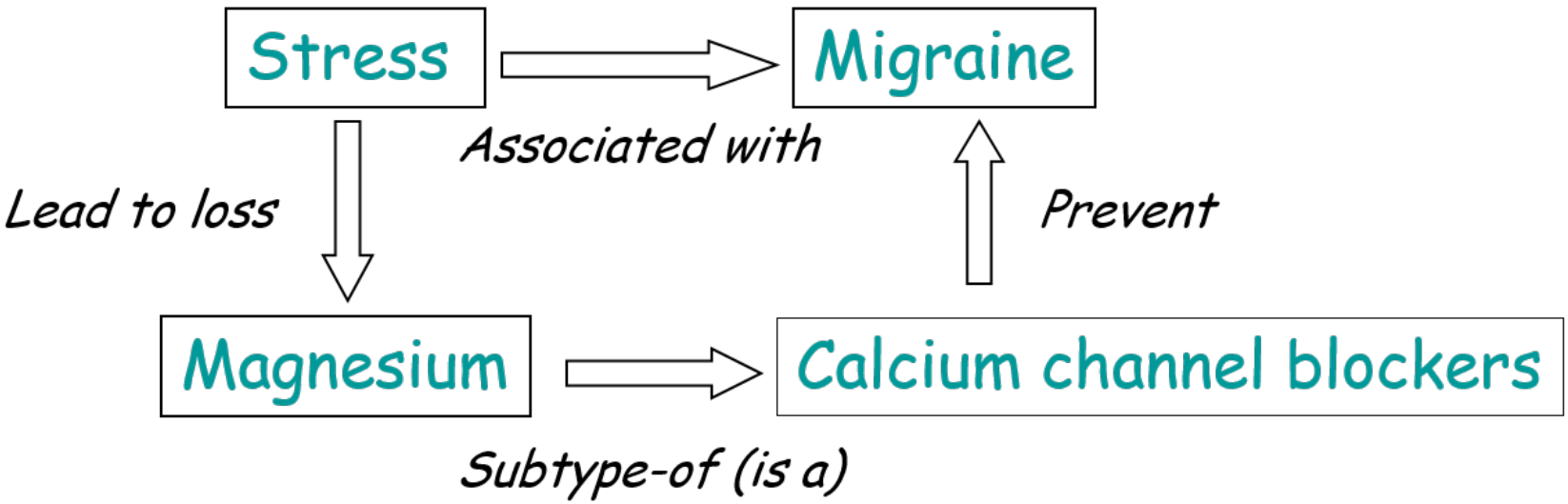
Text:

- Stress is associated with migraines
- Stress can lead to loss of magnesium
- Calcium channel blockers prevent some migraines
- Magnesium is a natural calcium channel blocker

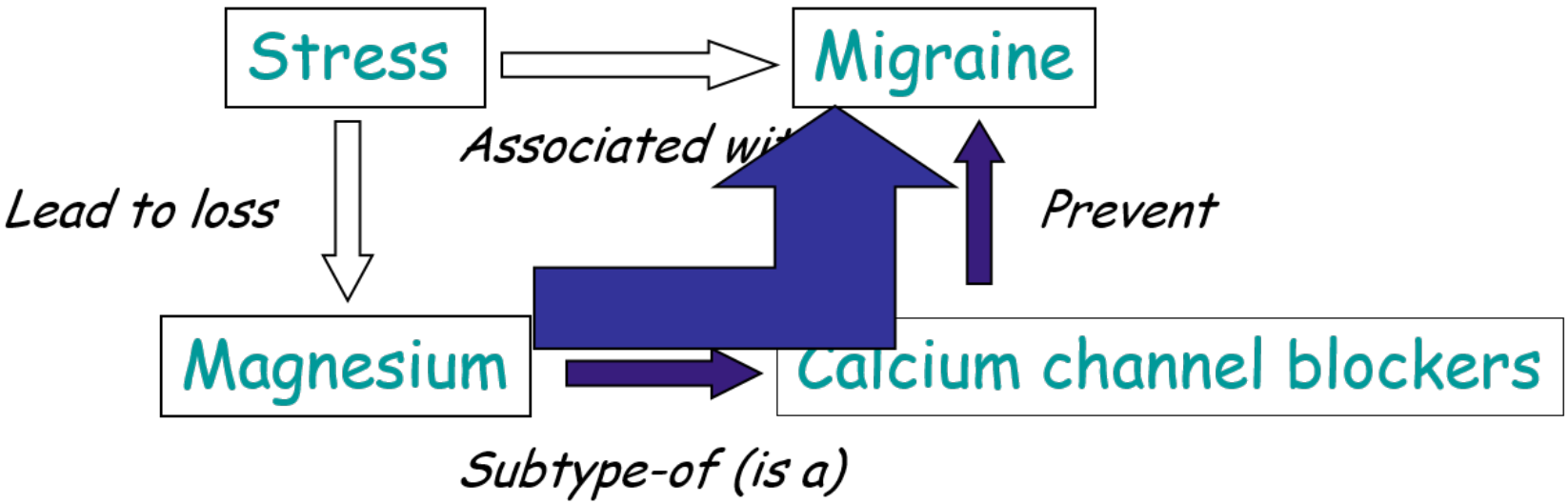
Extract semantic entities from text



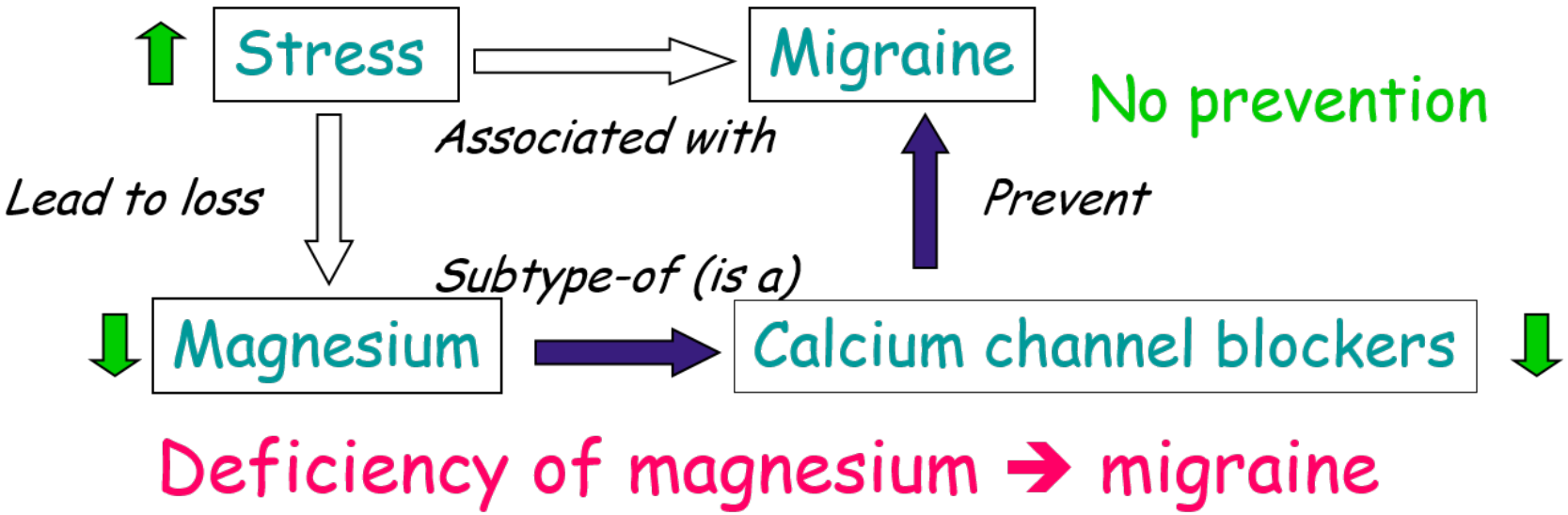
Classify relations between entities



Do reasoning: find new correlations



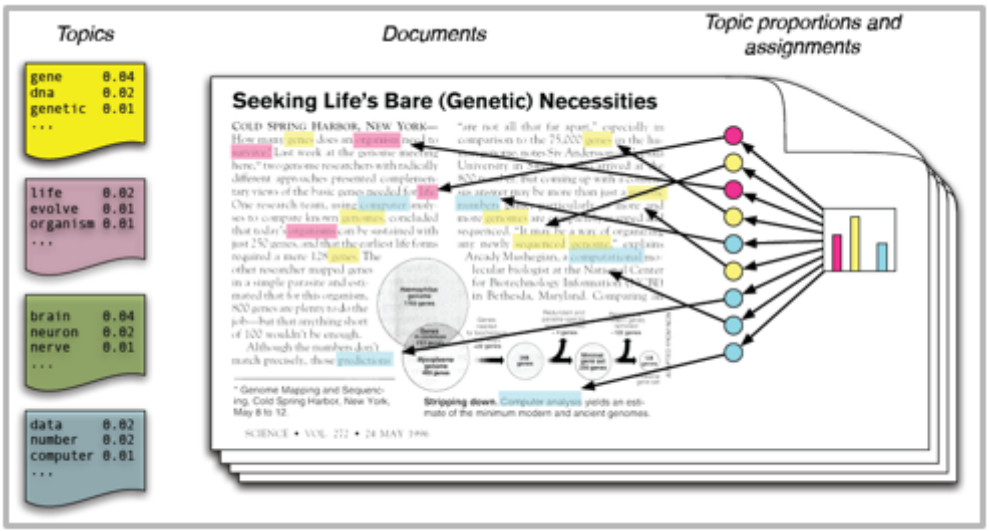
Do reasoning: infer causality



Topic Modeling

What is Topic Modeling?

- Topic modeling is a form of text mining, a way of identifying patterns in a corpus.
- You take your corpus and run it through a tool which groups words across the corpus into 'topics' (clusters of words) by a process of similarity.
- In a good topic model, the words in topic make sense, for example "navy, ship, captain" and "tobacco, farm, crops."



Core Concept: Similarity

Are these statements similar?

- "A seven-year quest to collect samples from the solar system's formation ended in triumph in a dark and wet Utah desert this weekend."
- "For a month, a huge storm with massive lightning has been raging on Jupiter under the watchful eye of an orbiting spacecraft."
- "One of Saturn's moons is spewing a giant plume of water vapour that is feeding the planet's rings, scientists say."

How would you quantify their similarity? How would you decide that two are more similar to each other than to the third?

Semantic Similarity

- A study done in Australia in 2005 : what a sample of Australian college students think is similar.
- Students read hundreds of paired short excerpts from news articles and ranked the pairwise similarity on a scale.
- Then examined all the classifications, and found that they had a correlation of 0.6.
- That's obviously a positive correlation, but not terribly high.
- So, humans doesn't necessarily agree with each other about semantic similarity
- It's kind of a fuzzy notion

Semantic Similarity: Any number?

- The study mentioned found that a particular topic modelling technique called latent semantic analysis (LSA) could achieve also a 0.6 correlation with the human ratings
- Correlating with the study participant's choice about as well as they correlated with each other.

Who cares about semantic similarity?

Some use cases:

- Query large collections of text: Traditionally used on large document collections. Legal discovery. Answer questions or aid searching huge corpora like government regulations, manuals, patent databases, etc.
- Automatic metadata: system can intelligently suggest tags and categories for documents based on other documents they're similar too.
- Recommendations: plagiarism detection, exam scoring. And there are a number of other use cases.
- Better human-computer interaction: Matching on similarity rather than words or with regexes allows us to accept broader ranges of input, in theory.

What is Topic Modeling?

- Topic modelling attempts to uncover the underlying semantic structure of text (or other data) by using statistical techniques to identify abstract, recurring patterns of terms in a set of data.
- These patterns are called topics. They may or may not correspond to our intuitive notion of a topic.
- Topic modelling models documents as collections of features, representing the documents as long vectors that indicate the presence/absence of important features, for example, the presence or absence of words in a document.
- We can use those vectors to create spaces and plot the locations of documents in those spaces and use that as a kind of proxy for their meaning.

What is Topic Modeling?

- Topic modeling is an unsupervised algorithm.
- One of the key ideas behind it is that every topic is present to varying degrees within each document.
- The typical output of running a corpus through TM is a set of distinctive words for each topic and for each document a percentage value indicating the presence of each topic within that document.
- One of the popular algorithms underlying TM is called Latent Dirichlet Allocation (LDA) and was described in a paper published by Blei, Ng and Jordan in 2003.

What Topic Modeling is NOT?

Topic modelling:

- Does not parse sentences.
- In fact, knows nothing about word order.
- Makes no attempt to "understand" grammar or language syntax.

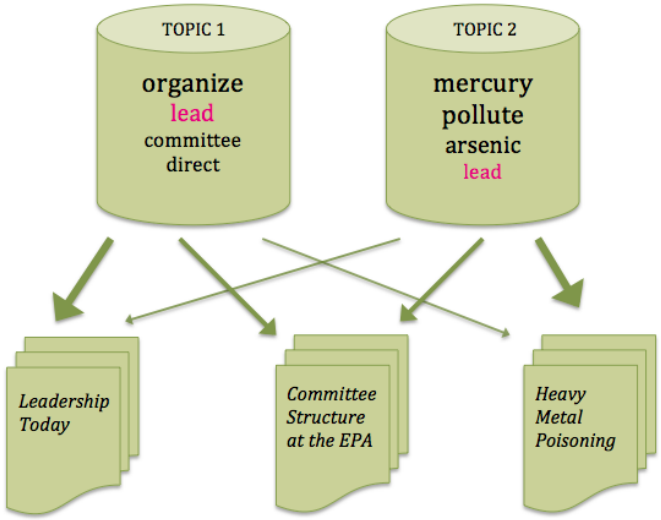
How does it work?

Manual mode:

- Imagine working through an article with a set of highlighters
- A different color for the key words of themes within the paper as you come across them
- Once done, you could copy out the words as grouped by the color you assigned them. That list of words is a topic, and each color represents a different topic.
- The popular LDA which is used to extract topics is based on "Dirichlet Distribution".
- Trying to understand LDA (and its proof) is not trivial (so leave it!!)

Still, how does it work?

- Each document contains a mixture of different topics.
- "topic" can be understood as a collection of words that have different probabilities of appearance.
- One topic might contain many occurrences of "organize," "committee," "direct," and "lead."
- Another might contain a lot of "mercury" and "arsenic," with a few occurrences of "lead."



Still, how does it work?

- Can't directly observe topics; what we have are documents
- Topic modeling is a way of extrapolating backward to infer the discourses ("topics") that could have generated them.
- How are we going to decide whether this occurrence of W belongs to topic Z?
- But one way to guess is to consider two questions.
  - How often does "lead" appear in topic Z?
  - How common is topic Z in the rest of this document?
- We will find probability of W present in Z for this document D, as follows:

$$P(Z|W,D) = \frac{\#\ of\ word\ W\ in\ topic\ Z + \beta_w}{total\ tokens\ in\ Z + \beta} * (\#\ words\ in\ D\ that\ belong\ to\ Z + \alpha)$$

- Its is calculated by multiplying the frequency of this word type W in Z by the number of other words in document D that already belong to Z.

How Topics look?

```
>>> model.show_topics()
'-0.203*"smith" + 0.166*"jan" + 0.132*"soccer" + 0.132*"software" +
     0.119*"fort" + -0.119*"nov" + 0.116*"miss" + -0.114*"opera" +
     -0.112*"oct" + -0.105*"water"',

'0.179*"squadron" + 0.158*"smith" + -0.140*"creek" + 0.135*"chess" +
     -0.130*"air" + 0.128*"en" + -0.122*"nov" + -0.120*"fr" + 0.119*"jan" +
     -0.115*"wales"',

'0.373*"jan" + -0.236*"chess" + -0.234*"nov" + -0.208*"oct" + 0.151*"dec" +
     -0.106*"pennsylvania" + 0.096*"view" + -0.092*"fort" + -0.091*"feb" +
     -0.090*"engineering"',
```

- These three abbreviated topics were extracted from a large corpora of texts by gensim.
- Words are not ordered.
- Positive and negative weights for each word, which get smaller in magnitude as the topic goes on.

Example

Latent Dirichlet allocation (LDA) is a technique that automatically discovers topics. It achieves the above results in 3 steps.

- Step 1: You tell the algorithm how many topics you think there are. Its just a number. Then topics would have ID's like Topic1, Topic2, etc.
- Step 2: Assign every word to a temporary topic.
- Step 3: In iteration, for each doc, for each word, its topic assignment is updated based on two criteria:
  - How prevalent is that word across topics?
  - How prevalent are topics in the document?

Example

- Following shows two documents X and Y, and number of topics to be be generated is 2.
- Lets call them Topics F (for Food) and P (for Pets)

Document X	Document Y
Fish	Fish
Fish	Fish
Eat	Milk
Eat	Kitten
Vegetables	Kitten

Example

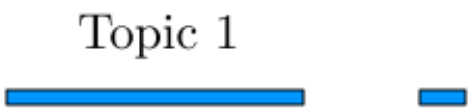
- Go to every document and for each word write its Topic ID, RANDOMLY.
- Say, in the updating iteration you need to update Topic for the first word of Doc Y: “fish” in Doc Y.
- Rest all the topic-word-docs assignments, although RANDOM are ASSUMED to be correct (this method/sampling is called Gibb’s sampling)
- How to compute that?

	Document X		Document Y
F	Fish	?	Fish
F	Fish	F	Fish
F	Eat	F	Milk
F	Eat	P	Kitten
F	Vegetables	P	Kitten

How prevalent is that word across topics?

- In Doc X, “fish” is with TopicF in both the cases, ie 2/2. It is with TopicP 0 times.
- In (remaining) Doc Y, “fish” is with TopicF in 1 case, ie 1/1. It is with TopicP 0 times.
- Total: “fish” with TopicF 3/3 times and with TopcP 0 times.

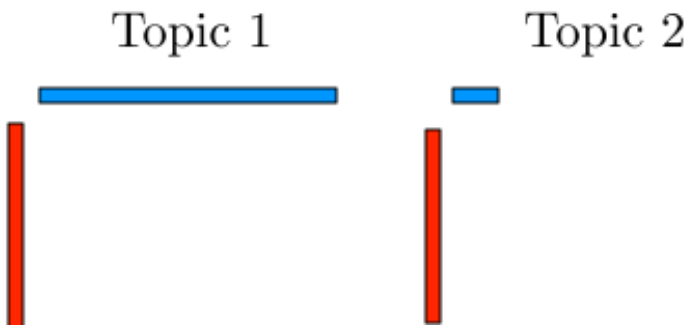
	Document X		Document Y
F	**Fish**	?	Fish
F	**Fish**	**F**	**Fish**
F	Eat	F	Milk
F	Eat	P	Kitten
F	Vegetables	P	Kitten



How prevalent are topics in the document?

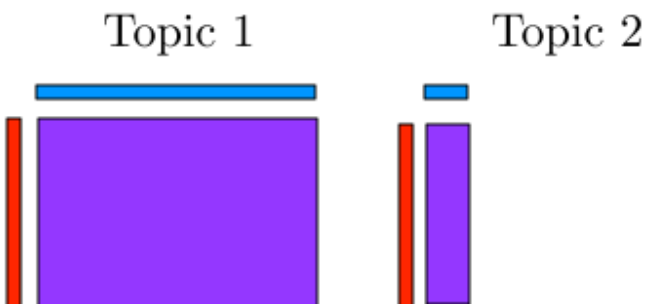
- Now within the single document, ie Doc Y, how are topics distributed?
- 2 TopicFs, 2 TopicPs. 50-50%
- So, their length bars are equal (but vertical!!)

	Document X		Document Y
F	Fish	?	Fish
F	Fish	F	Fish
F	Eat	F	Milk
F	Eat	P	Kitten
F	Vegetables	P	Kitten



Probability of Assignment

- Multiply horizontal and vertical lengths, get get areas



- Topic1 has more area, so “fish” going to TopcF is proportionally more
- So, update the assignment value of “fish” in DOc Y to be “TopicF”.
- Repeating this ‘n’ iterations, words cluster to topics and topics cluster to documents.

Interesting fact

- Even though value for TopicP was 0, we did not show the bar length to be 0, but some value.
- They are hyper parameters, one in each dimension, alpha and beta.
- alpha is for value adding to word-topics say 0.8 as smoothing constant
- beta is for value added to doc-topics likings

Results

- In conclusion, we would assign the "fish" word of Doc Y to Topic F. Go to next word (assume remaining assignments are perfect)
- The process of checking topic assignment is repeated for each word in every document, cycling through the entire collection of documents multiple times.
- This iterative updating is the key feature of LDA that generates a final solution with coherent topics.

Named Entity Recognition

Introduction

In NLP, NER is a method of extracting the relevant information from a large corpus and classifying those entities into predefined categories such as location, organization, name and so on.

When Sebastian Thrun PERSON started at Google ORG in 2007 DATE, few people outside of the company took him seriously. "I can tell you very senior CEOs of major American NORP car companies would shake my hand and turn away because I wasn't worth talking to," said Thrun PERSON, now the co-founder and CEO of online higher education startup Udacity, in an interview with Recode ORG earlier this week DATE.

A little less than a decade later DATE, dozens of self-driving startups have cropped up while automakers around the world clamor, wallet in hand, to secure their place in the fast-moving world of fully automated transportation.

(Ref: Complete Tutorial on Named Entity Recognition (NER) using Python and Keras - Akshay Chavan)

The who, where, when & how much in a sentence

The task: identify atomic elements of information in text

- Person names
- Company/organization names
- Locations
- Dates & times
- Percentages
- Monetary amounts

The decision by the independent MP Andrew Wilkie to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the 2010 election, Wilkie, Rob Oakeshott, Tony Windsor and the Greens agreed to support Labor, they gave just two guarantees: confidence and supply.

Person
Date
Location
Organi-
zation

Example

Labels	Labels Meaning	Example
geo	Geographical Entity	Dubai
org	Organization	New York Times
per	Person	Harry
gpe	Geopolitical Entity	European
tim	Time Indicator	Friday
art	Artifact	Economics
eve	Event	Olympic
nat	Natural Phenomenon	Hurricane
O	Other	of, an, the

(Ref: Complete Tutorial on Named Entity Recognition (NER) using Python and Keras - Akshay Chavan)

Difficulties

- Too numerous to include in dictionaries
- Changing constantly: new names invent unknown words
- Appear in many variant forms: e.g. John Smith, Mr Smith, John
- Subsequent occurrences might be abbreviated
- List search/matching does't perform well

Concept

Whether a phrase is a proper name, and what name class it has, depends on

- Internal structure:“Mr. London”
- Context:“The new location, London, will make a better place.”

Applications

- Information Extraction: relations are associations between named entities
- Named entities can be indexed, linked off, etc.
- Sentiment can be attributed to companies or products
- Summary generation
- Machine Translation
- Document organization/classification
- Increase accuracy of Internet search results (location Clinton/South Carolina vs. President Clinton)
- For question answering, answers are often named entities

Standard Approaches

- Hand-written regular expressions: Perhaps stacked
- Using classifiers: Naïve Bayes, Maxent models
- Sequence models:HMMs, CRFs

Traditional	Statistical	Neural
• Rules • Regular Expressions • Gazetteers	• Word-based models – PMI, log-likelihood. • Sequence models – Conditional Random Fields	• Bi-LSTM • Bi-LSTM+CRF • Transformer based Models

Note: Gazetteers are dictionaries or lookup lists.

(Ref: Sujit Pal: Building Named Entity Recognition Models Efficiently Using NERDS — PyData LA 2019)

The hand-crafted/Rule-based approach

Uses hand-written context-sensitive reduction rules:

- For certain restricted, common types of entities in unstructured text, simple regex patterns also usually work.
- Finding (US) phone numbers
- `(?:\(?[0-9]{3}\)?[ -.])?[0-9]{3}[ -.]?[0-9]{4}`
- Title capitalized word : title person_name compare “Mr. Jones” vs. “Mr. Ten-Percent” : no rule without exceptions
- person_name, “the” adj* “CEO of” organization “Fred Smith, the young dynamic CEO of BlubbCo” : ability to grasp non-local patterns

Plus help from databases of known named entities

Methods for Sequence Labeling

Typically the following methods are used for NER:

- Hidden Markov Model (HMM)
- Maximum Entropy Classifier (MaxEnt)
- Maximum Entropy Markov Model (MEMM)
- Conditional Random Fields (CRF)

These are all classifiers (i.e., supervised learning) which model sequences (rather than individual random variables)

Sequence approach

Training

- Collect a set of representative training documents
- Label each token for its entity class or other (O)
- Design feature extractors appropriate to the text and classes 3. Design feature extractors appropriate to the text and classes
- Train a sequence classifier to predict the labels from the data

Testing

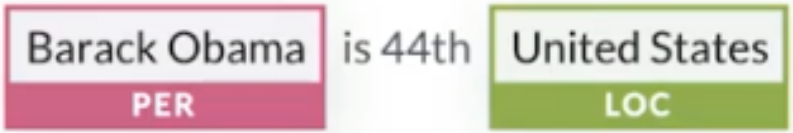
- Receive a set of testing documents
- Run sequence model inference to label each token
- Appropriately output the recognized entities

Features for sequence labeling

- Words:
  - Current word (essentially like a learned dictionary)
  - Previous/next word (context)
- Other kinds of inferred linguistic classification: Part-of-speech tags
- Label context: Previous (and perhaps next) label
- Word Shapes: Map words to simplified representation that encodes attributes such as length, capitalization, numerals, Greek letters, internal punctuation, etc.

Varicella-zoster	Xx-xxx
mRNA	xXXX
CPA1	XXXd

BIO or IOB format



- **BIO** – **B**egin **I**n **O**ut.
  - Barack/**B-PER** Obama/**I-PER** is/O 44th/O United/**B-LOC** States/**I-LOC** President/O ./O
- **BILOU** – a tagging variant:
  - **U** – **Unit** token (for single token entities)
  - **L** – **Last** token in sequence, ex. Barack/B-PER Obama/L-PER

CoNLL

Barack	B-PER
Obama	I-PER
is	O
44th	O
United	B-LOC
States	I-LOC
President	O
.	O

(Ref: Sujit Pal: Building Named Entity Recognition Models Efficiently Using NERDS — PyData LA 2019)

Features and Output Data format

	IO encoding	IOB encoding
Fred	PER	B-PER
showed	O	O
Sue	PER	B-PER
Mengqiu	PER	B-PER
Huang	PER	I-PER
's	O	O
new	O	O
painting	O	O

VBG	NN	IN	DT	NN	IN	NN
Chasing	opportunity	in	an	age	of	upheaval

POS tagging

PERS	O	O	O	ORG	ORG
Murdoch	discusses	future	of	News	Corp.

Named entity recognition

B	B	I	I	B	I	B	I	B	B
而	相	对	于	这	些	品	牌	的	价

Word segmentation

Text segmentation

Dataset example

CoNLL-2003

- CoNLL - Conference on Natural Language Learning by the ACL's Special Interest Group on Natural Language Learning
- Shared Task: Language-Independent Named Entity Recognition
- Goal: Identify boundaries and types of named entities

Word	POS	Chunk	EntityType
U.N.	NNP	I-NP	I-ORG
official	NN	I-NP	O
Ekeus	NNP	I-NP	I-PER
heads	VBZ	I-VP	O
for	IN	I-PP	O
Baghdad	NNP	I-NP	I-LOC
.	.	O	O

- Inputs: $x = (x_1, \ldots, x_n)$
- Labels: $y = (y_1, \ldots, y_n)$
- Typical goal: Given x, predict y

Algorithms

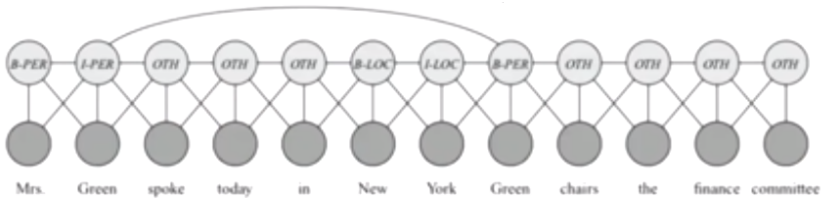
Conditional Random Fields (CRFs)

- CRFs are used for predicting the sequences that use the contextual information to add information which will be used by the model to make a correct prediction.
- The output sequence is modeled as the normalized product of the feature function.
- Actually its similar to basic Logistic Regression, done for each token, with a custom feature lists, having info of the context.

$$p(y|X,\lambda) = \frac{1}{Z(X)} exp \sum_{i=1}^{n} \sum_{j} \lambda_j f_j(X, i, y_{i-1}, y_i)$$

Task specific predictions

- Inputs: words (e.g., Bill, Microsoft, etc)
- Outputs: Tags (e.g. PERSON, ORG, etc)
- Features: context (neighboring words, POS, etc)
- Why Probabilistic Graphical Model like CRF? : features are correlated. Not like Naive Bayes.
- So, instead of modeling JOINT distribution of X and Y, CRF is trying to model CONDITIONAL distribution of Y given X. So, correlations amongst Xs are irrelevant here.



Features: word capitalized, word in atlas or name list, previous word is "Mrs", next word is "Times", ...

(Ref: Conditional Random Fields - Stanford University - Daphne Koller)

Feature Preparation for CRF

In [1]:
```
def word2features(sent, i):
    word = sent[i][0]
    postag = sent[i][1]

    features = {
        'bias': 1.0,
        'word.lower()': word.lower(),
        'word[-3:]': word[-3:],
        'word[-2:]': word[-2:],
        'word.isupper()': word.isupper(),
        'word.istitle()': word.istitle(),
        'word.isdigit()': word.isdigit(),
        'postag': postag,
        'postag[:2]': postag[:2],
    }
    if i > 0:
        word1 = sent[i-1][0]
        postag1 = sent[i-1][1]
        features.update({
            '-1:word.lower()': word1.lower(),
            '-1:word.istitle()': word1.istitle(),
            '-1:word.isupper()': word1.isupper(),
            '-1:postag': postag1,
            '-1:postag[:2]': postag1[:2],
        })
    else:
        features['BOS'] = True

    if i < len(sent)-1:
        word1 = sent[i+1][0]
        postag1 = sent[i+1][1]
        features.update({
            '+1:word.lower()': word1.lower(),
            '+1:word.istitle()': word1.istitle(),
            '+1:word.isupper()': word1.isupper(),
            '+1:postag': postag1,
            '+1:postag[:2]': postag1[:2],
        })
    else:
        features['EOS'] = True

    return features


def sent2features(sent):
    return [word2features(sent, i) for i in range(len(sent))]

def sent2labels(sent):
    return [label for token, postag, label in sent]

def sent2tokens(sent):
    return [token for token, postag, label in sent]
```

Training the model with scikit-learn

In [15]:
```
crf = CRF(algorithm = 'lbfgs',
          c1 = 0.1,
          c2 = 0.1,
          max_iterations = 100,
          all_possible_transitions = False)
crf.fit(X_train, y_train)
```

Out[15]:
```
CRF(algorithm='lbfgs', all_possible_states=None,
  all_possible_transitions=False, averaging=None, c=None, c1=0.1, c2=0.1,
  calibration_candidates=None, calibration_eta=None,
  calibration_max_trials=None, calibration_rate=None,
  calibration_samples=None, delta=None, epsilon=None, error_sensitive=None,
  gamma=None, keep_tempfiles=None, linesearch=None, max_iterations=100,
  max_linesearch=None, min_freq=None, model_filename=None,
  num_memories=None, pa_type=None, period=None, trainer_cls=None,
  variance=None, verbose=False)
```

Evaluating the model performance

In [20]:
```
report = flat_classification_report(y_test, y_pred)
print(report)
```

	precision	recall	f1-score	support
B-art	0.43	0.17	0.24	78
B-eve	0.68	0.41	0.51	61
B-geo	0.86	0.91	0.88	7481
B-gpe	0.97	0.94	0.95	3185
B-nat	0.85	0.36	0.51	47
B-org	0.81	0.74	0.77	4187
B-per	0.86	0.83	0.84	3421
B-tim	0.93	0.88	0.90	4030
I-art	0.25	0.08	0.12	64
I-eve	0.52	0.30	0.38	44
I-geo	0.81	0.80	0.80	1461
I-gpe	0.81	0.46	0.59	37
I-nat	0.50	0.17	0.25	12
I-org	0.83	0.81	0.82	3441
I-per	0.86	0.90	0.88	3488
I-tim	0.84	0.74	0.79	1245
O	0.99	0.99	0.99	177951
avg / total	0.97	0.97	0.97	210233

Summary

- CRF is parametrized the same as a Gibbs distribution but normalized differently.
- Generalizes Logistic Regression Model.

(Ref: Conditional Random Fields - Stanford University - Daphne Koller)

Conclusions

Hand-crafted vs. Automated

Hand-made systems:

- Can achieve higher performance than ML systems
- Non-local phenomena best handled by regular expressions
- Several person-months for rule-writing, requires experienced linguists
- Rules depend on specific properties of language, domain & text format
- Manual adaption necessary when domain changes
- Re-write rules for other languages

Hand-crafted vs. Automated

Automated approaches:

- Train on human-annotated texts
- No expensive computational linguists needed
- 1,00,000 words can be tagged in 1-3 days
- Ideally, no manual work required for domain changes
- Easier to port to other languages
- Features are locally limited

Evaluation for NER

- Recall and precision are straightforward for tasks like IR and text categorization, where there is only one grain size (documents)
- The measure behaves a bit funnily for IE/NER when there are boundary errors (which are common):
- First Bank of Chicago announced earnings
- This counts as both a FP and a FN
- Selecting nothing would have been better
- Some other metrics (e.g., MUC scorer) give partial credit (according to complex rules)

References

Many publicly available resources have been refereed for making this presentation. Some of the notable ones are:

- Nltk - Steven Bird, Ewan Klein, Edward Loper
- Machine Learning for Natural Language Processing - Traian Rebedea, Stefan Ruseti - LeMAS 2016 - Summer School
- Natural Language Processing with Python - District Data Labs
- Natural Language Processing+ Python - Ann C. Tan-Pohlmann
- Applied Natural Language Processing - Barbara Rosario
- Named Entity Recognition - Stephan Lesch
- Galvanize NLP
- Text Mining - Behrang QasemiZadeh
- Natural Language Processing - Information Extractoin, Christopher Manning
- Topic Modeling - (William Bert, Megan R Brett, Ted Underwood, Algobeans, Shivam Bansal)
- Word2Vec - (Girish K.,Sivan Biham & Adam Yaari, Adrian Colyer)

Thanks ... yogeshkulkarni@yahoo.com