Chapter IR:III

III. Text Transformation

- ❑ Text Statistics
- ❑ Parsing Documents
- ❑ Information Extraction
- ❑ Link Analysis

Information Extraction

Overview

Information extraction is the task of extracting structured information from unstructured information sources.

Key goals:

- Rendering text amenable to structured processing and retrieval.
- Allowing for logical reasoning and inferences.

Basic analysis process:

- Input: Token stream
- Output: Token stream with annotations / tags on (subsequences of) tokens.

Methodology:

- Extraction rules
- Machine learning
- Sequence modeling

Information Extraction

Overview

Information extraction is employed to identify more complex index terms by means of natural language processing technology (computationally expensive):

- Noun phrases

  Phrases which have a noun as its head word, i.e., a noun and any word that modifies it. Examples: “*The yellow house* is for sale.”, “I want *a skate board*”.

- Named entities

  Words or phrases that designate something (e.g., a place, a person, an organization, etc.).

- Coreference resolution

  Coreferences, i.e., anaphora and cataphora, are expressions that refer backward or forward in a text, respectively. Resolving *them* is important for text understanding, yet, one of the most difficult problems of natural language processing

- Relation detection

  Extraction of relations between named entities mentioned in the text. Example: “*Bill* lives in the *USA*”.

- Semi-structured information extraction

  Extraction of tables, quotes, references, comments, etc.

Information Extraction

Part-of-Speech Tagging

Given a token sequence of text, markup each token with its part of speech (POS).

Common English (Western) parts of speech:

- Noun (names of abstract or concrete entities: persons, places, things, ideas, qualities)
- Pronoun (substitutes for nouns)
- Verb (actions, occurrences, or states of being)
- Adjective (modifiers of a noun or pronoun)
- Adverb (modifiers of verbs, adverbs, or adjectives)
- Preposition (words expressing relations in a phrase or sentence)
- Conjunction (connects words, phrases, or clauses)
- Interjection (expressions of feelings and emotions)
- Determiner (markers of definiteness or indefiniteness)

For practical purposes, these broad classes are insufficient. Typically some 30 to 150 parts of speech are distinguished.

Information Extraction

Part-of-Speech Tagging: Example

Original text:

```
A relevant document will describe marketing strategies carried out by
U.S. companies for their agricultural chemicals, report predictions for
market share of such chemicals, or report market statistics for
agrochemicals, pesticide, herbicide, fungicide, insecticide, fertilizer,
predicted sales, market share, stimulate demand, price cut, volume of
sales.
```

Brill tagger:

A/**DT** relevant/**JJ** document/**NN** will/**MD** describe/**VB** marketing/**NN** strategies/**NNS** carried/**VBD** out/**IN** by/**IN** U.S./**NNP** companies/**NNS** for/**IN** their/**PRP$** agricultural/**JJ** chemicals/**NNS** ,/, report/**NN** predictions/**NNS** for/**IN** market/**NN** share/**NN** of/**IN** such/**JJ** chemicals/**NNS** ,/, or/**CC** report/**NN** market/**NN** statistics/**NNS** for/**IN** agrochemicals/**NNS** ,/, pesticide/**NN** ,/, herbicide/**NN** ,/, fungicide/**NN** ,/, insecticide/**NN** ,/, fertilizer/**NN** ,/, predicted/**VBN** sales/**NNS** ,/, market/**NN** share/**NN** ,/, stimulate/**VB** demand/**NN** ,/, price/**NN** cut/**NN** ,/, volume/**NN** of/**IN** sales/**NNS** ./.

CC	coordinating conjunction
DT	singular determiner/quantifier
IN	preposition
JJ	adjective
MD	modal auxiliary
NN	singular or mass noun
NNP	proper noun, singular
NNS	plural noun
PRP$	possessive pronoun
VB	verb, base form
VBD	verb, past tense
VBN	verb, past participle
,	comma
.	dot
	other tags

Information Extraction

Part-of-Speech Tagging: Brill Tagger [Brill 1992]

Principle: “error-driven transformation-based tagging”

1. Assign each token its most likely part of speech tag. Stemming rules are applied to match inflected tokens with word stems stored in a dictionary.
2. Apply a list of transformation rules to correct tagging errors.
3. Repeat Step 2 until no rules can be applied, anymore, or after a pre-specified number of repetitions.

Concepts:

- Initial tag probabilities are trained on a large pre-tagged corpus.
- Rules are learned from errors made on a pre-tagged corpus, and applied in the order listed.
- Rules are defined as follows: `T1 T2 <Premise>`

Semantics:

For each token currently tagged with `T1` which fulfills the `<Premise>`, replace `T1` with `T2`.

Information Extraction

Part-of-Speech Tagging: Brill Tagger [Brill 1992]

Premises:

`context x`	A word in `context` is tagged `x`.
`property`	The word has a certain `property`.
`context property`	A word in `context` has a certain `property`.
`context`	One or any of $i \in [1, 3]$ preceding or following word(s).
`property TRUE|FALSE`	Capitalized word.

Example rules:

`TO`	`IN`	`next-tag AT`	`TO`: `to`, `IN`: preposition, `AT`: article
`VBN`	`VBD`	`prev-word-is-cap TRUE`	`VBN`: verb, past participle,
`VBD`	`VBN`	`prev-1-or-2-or-3-tag HVD`	`VBD`: verb, past tense, `HVD`: auxiliary had
`VB`	`NN`	`prev-1-or-2-tag AT`	`VB`: verb, base form
`NN`	`VB`	`prev-tag TO`	`NN`: noun, singular or mass
`TO`	`IN`	`next-word-is-cap TRUE`	
`NN`	`VB`	`prev-tag MD`	`MD`: modal

Rules are learned starting with the initial tagging on a training dataset by instantiating rules from the above templates, keeping those that minimize tagging errors the most in each iteration, until some termination criterion is reached.

Information Extraction

Part-of-Speech Tagging: Brill Tagger [Brill 1994]

Problem: The tagger cannot tag words not occurring in the training data.

An unknown word tagger can be trained based on the same principles but with different premises as templates for rules. `T1` may be `UNK` for unknown.

Premises:

`affix x constraint`	Token fulfills `constraint` regarding affix of at most 4 chars.
`context word`	A word appears in `context`.
`char x`	Character `x` occurs in word.
`constraint`	When deleting or adding affix `x`, word found in dictionary. Else, affix `x` occurs in token.

Example rules:

`NN`	`NNS`	`suffix -s occurs`	`NN`: noun, singular or mass, `NNS`: noun, plural
`NN`	`CD`	`char .`	`CD`: cardinal number
`NN`	`JJ`	`char -`	`JJ`: adjective
`NN`	`VBN`	`suffix -ed occurs`	`VBN`: verb, past participle
`NN`	`VBG`	`suffix -in occurs`	`VBG`: verb, gerund or present participle
`UNK`	`JJ`	`suffix -ly addition`	`UNK`: unknown
`UNK`	`RB`	`suffix -ly occurs`	`RB`: adverb

Remarks:

- Large corpora for part of speech tagging have been painstakingly manually annotated, starting with the 1 million word Brown corpus in the 1960s, later superseded by the 100 million word British National Corpus, and others.

- Tag sets: Brown (87 tags), Penn TreeBank II (41 tags), British National Corpus (61 tags), Oxford English part-of-speech tagset (109 tags), British National Corpus Sampler (146 tags).

- Assigning the most probable tag to each known word and proper noun (`NNP`) to all unknown words already yields 90% accuracy. [Charniak 1997]

- The state of the art in part of speech tagging can be reviewed at NLP-progress; an outdated overview is found at aclweb.org. Most taggers reported are based on statistical sequence models rather than rules. However, many taggers proposed are not included, including the Brill tagger.

- Nevertheless, the Brill tagger frequently serves as baseline for comparison, and as a last step in tagging pipelines.

Information Extraction

Noun Phrase

A phrase is a group of words (or possibly a single word) that functions as a constituent in the syntax of a sentence. A noun phrase (NP) is a phrase which has a noun (or indefinite pronoun) as its head word, preceded and/or followed by modifying words. Hence, a noun phrase is a phrase which acts like a noun.

Common noun modifiers:

- Determiners / articles (the inclusion of determiners is disputed)
- Adjectives
- Prepositional phrases

Examples: What are noun phrases in the examples?

- Cats sleep a lot.
- A cat is sleeping.
- The fluffy, long-haired cat sleeps.
- The cat on top of the stool is sleeping.
- Most big cats sleep at daytime.

Information Extraction

Noun Phrase

A phrase is a group of words (or possibly a single word) that functions as a constituent in the syntax of a sentence. A noun phrase (NP) is a phrase which has a noun (or indefinite pronoun) as its head word, preceded and/or followed by modifying words. Hence, a noun phrase is a phrase which acts like a noun.

Common noun modifiers:

- Determiners / articles (the inclusion of determiners is disputed)
- Adjectives
- Prepositional phrases

Examples:

- *Cats* sleep a lot.
- A *cat* is sleeping.
- The fluffy, long-haired *cat* sleeps.
- The *cat* on top of the stool is sleeping.
- Most big *cats* sleep at daytime.

Information Extraction

Noun Phrase Extraction

Noun phrases can be extracted from POS tagged text using regular expressions.

To extract noun phrases as index terms, simplified patterns are typically used:

- sequences of nouns
- sequences of adjectives followed by nouns

Example (noun phrases bold, tagging errors fixed):

A/DT **relevant/JJ document/NN** will/MD describe/VB **marketing/NN strategies/NNS** carried/VBD out/IN by/IN **U.S./NNP companies/NNS** for/IN their/PRP$ **agricultural/JJ chemicals/NNS** ,/, report/VB predictions/NNS for/IN **market/NN share/NN** of/IN such/DT chemicals/NNS ,/, or/CC report/VB **market/NN statistics/NNS** for/IN agrochemicals/NNS ,/, pesticide/NN ,/, herbicide/NN ,/, fungicide/NN ,/, insecticide/NN ,/, fertilizer/NN ,/, **predicted/JJ sales/NNS** ,/, **market/NN share/NN** ,/, stimulate/VB demand/NN ,/, **price/NN cut/NN** ,/, volume/NN of/IN sales/NNS ./.

Compared to simply indexing n-grams, extracting noun phrases incurs high processing costs for POS tagging while offering the potential to save storage space for n-grams that would never be requested.

Information Extraction

Named Entity

A named entity refers to a real-world object that can be denoted with a proper name.

Named entities are phrases that form a subset of noun phrases, adjective phrases, or take the role of a noun phrase but have different internal structure.

Three basic object categories:

- Names of people, organizations, locations, facilities, products, events, natural objects, diseases, colors
- Times
- Numbers

Example: What are the named entities?

```
Fred Smith, living at 10 Water Street, Springfield, MA, has been breeding
five species of tropical fish for the past 15 years.
```

Information Extraction

Named Entity

A named entity refers to a real-world object that can be denoted with a proper name.

Named entities are phrases that form a subset of noun phrases, adjective phrases, or take the role of a noun phrase but have different internal structure.

Three basic object categories:

- Names of people, organizations, locations, facilities, products, events, natural objects, diseases, colors
- Times
- Numbers

Example:

`Fred Smith, living at 10 Water Street, Springfield, MA, has been breeding five species of tropical fish for the past 15 years.`

Information Extraction

Named Entity Recognition

Approaches to construct a named entity recognizer:

- **Dictionary-based recognition**
  Compile lists of words for each named entity category. If a (sequence of) token in a text is found in a text, or if it can be matched with a word from the dictionary, it is tagged as a named entity.
- **Rule-based recognition**
  Handcrafted or learned pattern rules are defined to recognize named entities. Some aspects of learning rules can be transferred from the Brill tagger.
- **Statistical sequence modeling**
  Application of probabilistic sequence models, such as hidden Markov models, to obtain named entities from sentences.

For every named entity category (or combinations thereof), a sufficiently large set of manually annotated training documents is required.

Domain transfer of trained recognizers between entity categories is difficult.

Information Extraction

Named Entity Recognition: Hidden Markov Models (informal)

Hypothesis:

- Whether a token is a named entity depends on tokens found around it.
  Example: `marathon` is part of a named entity when preceded by `Boston`.

➜ Modeling the sequence of token states (i.e., named entity, or not) may allow for recognizing them.

Sequence modeling with Markov models:

- A Markov model models a process (e.g., generating a sentence) as a collection of states with transition probabilities between them.
- If state transition probabilities depend only on the current state, the process modeled is said to have the Markov property.
- Markov assumption for text: the $i+1$st token in a sequence depends only on the ith token. Generalization: the last $i-n$ tokens.

➜ Problem: We want to know to what category of named entities a token belongs, not what the next most likely token is.

Information Extraction

Named Entity Recognition: Hidden Markov Models (informal)

Hypothesis:

- Whether a token is a named entity depends on tokens found around it.
  Example: `marathon` is part of a named entity when preceded by `Boston`.

➜ Modeling the sequence of token states (i.e., named entity, or not) may allow for recognizing them.

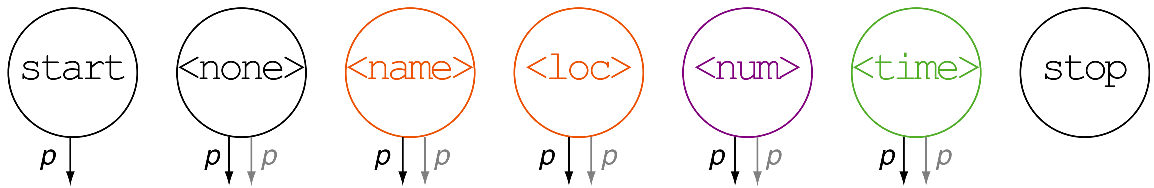
Sequence modeling with hidden Markov models:

- In a hidden Markov model, the state transition probabilities cannot be directly observed.
- Instead, every state may produce an output (e.g., a token from a vocabulary) with a certain probability.
- Based on examples, probabilities for state transitions and outputs are fitted so that the example texts are generated by the model with a high likelihood.

➜ Given a token sequence with unknown states, a trained model allows for inferring the hidden state transitions (e.g., named entity, or not).

Information Extraction

Named Entity Recognition: Hidden Markov Models (informal)



to any state except start with prior probability

output of word with probability

Example of hidden state transitions:

start $\xrightarrow{p}$ **<name>** $\xrightarrow{p}$ **<name>** $\xrightarrow{p}$ <none> $\xrightarrow{p}$ <none> $\xrightarrow{p}$ <none> $\xrightarrow{p}$ **<loc>** $\xrightarrow{p}$
<loc> $\xrightarrow{p}$ **<loc>** $\xrightarrow{p}$ **<loc>** $\xrightarrow{p}$ **<loc>** $\xrightarrow{p}$ **<loc>** $\xrightarrow{p}$ **<loc>** $\xrightarrow{p}$ <none> $\xrightarrow{p}$
<none> $\xrightarrow{p}$ <none> $\xrightarrow{p}$ <none> $\xrightarrow{p}$ **<num>** $\xrightarrow{p}$ **<num>** $\xrightarrow{p}$ **<num>** $\xrightarrow{p}$ **<num>** $\xrightarrow{p}$
<num> $\xrightarrow{p}$ <none> $\xrightarrow{p}$ **<time>** $\xrightarrow{p}$ **<time>** $\xrightarrow{p}$ **<time>** $\xrightarrow{p}$ **<time>** $\xrightarrow{p}$ stop

Example of observed / expected output (generated by each state with probability p):

Fred/**NNP** Smith/**NNP**,/, living/**VBG** at/**IN** 10/**CD** Water/**NNP** Street/**NNP**,/, Springfield/**NNP**,/, MA/**NNP**,/, has/**VBZ** been/**VBN** breeding/**VBG** five/**CD** species/**NNS** of/**IN** tropical/**JJ** fish/**NN** for/**IN** the/**DT** past/**JJ** 15/**CD** years/**NNS**./.

Information Extraction

Named Entity Recognition: Hidden Markov Models (informal)

Using dynamic programming algorithms, such as the Viterbi algorithm or the Forward-backward algorithm, the probabilities of hidden Markov models are trained.

Rule-based approaches are competitive to sequence modeling approaches, whereas performance depends on the category of named entity.

$> 90\%$ accuracy for names, locations, organizations require about 1 million tokens of training data (1500 news articles).

