Lecture 8

Text Mining

COMP 474/6741, Winter 2024

René Witte
Department of Computer Science
and Software Engineering
Concordia University

Outline

1. Introduction
2. Natural Language Processing (NLP)
3. Text Mining Applications
4. Notes and Further Reading

Slides Credit

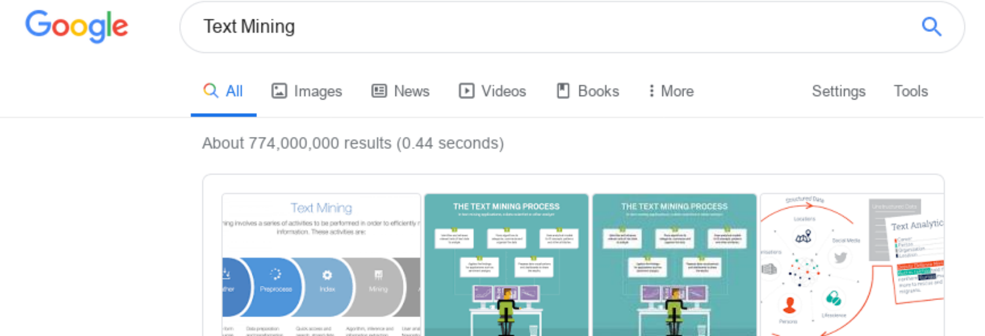
- Includes slides from Hoifung Poon, Chris Quirk & Scott Wen-Tau Yih, *Machine Reading for Precision Medicine*, https://www.microsoft.com/en-us/research/uploads/prod/2018/01/1802_aaai-tutorial_precision-med.pdf
- Includes slides from Matthew Honnibal & Ines Montani, *An introduction to spaCy*, https://github.com/explosion/talks/blob/master/2017-08-28_spaCy-101.pdf

Information Overload

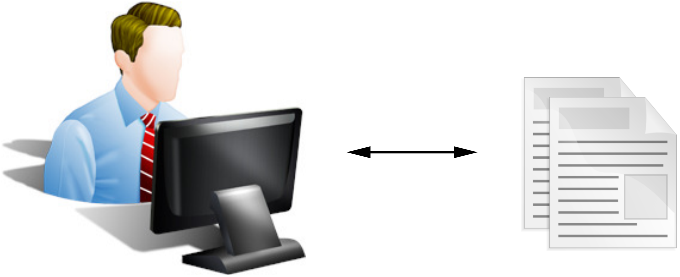
Too much (textual) information

- We now have electronic books, documents, web pages, emails, blogs, news, chats, memos, research papers, . . .
- . . . all of it immediately accessible, thanks to databases and Information Retrieval (IR)
- An estimated 80–85% of all data stored in databases are natural language texts
- But humans did not scale so well. . .

Results in the common perception of Information Overload (or even *Information Rage* when discussing the emotional impact)



Architecture 1.0



Knowledge Worker **Documents**

Example: Tumor Board KB Curation

The deletion mutation on exon-19 of EGFR gene was present in 16 patients, while the L858E point mutation on exon-21 was noted in 10.

All patients were treated with gefitinib and showed a partial response.

Gefitinib can treat tumors w. EGFR-L858E mutation

PubMed

27 million abstracts

Two new abstracts every minute

Adds over one million every year

Machine Reading

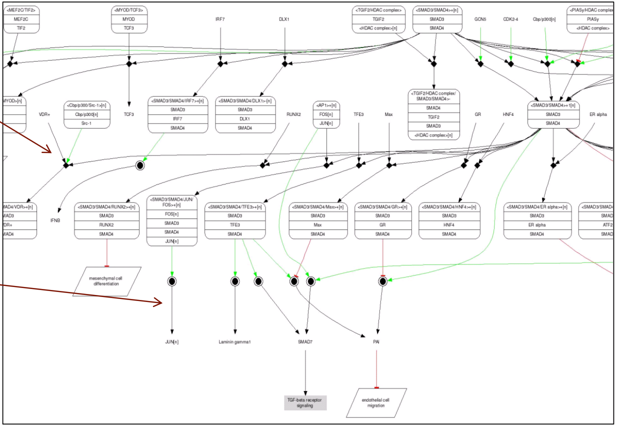
PMID: 123

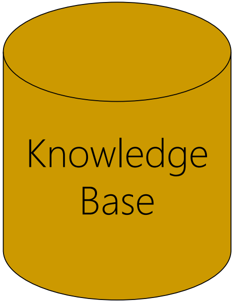
…
VDR+ binds to SMAD3 to form
…

PMID: 456

…
JUN expression is induced by SMAD3/4
…

⋮





Top Stories

UPDATED EVERY 10 MINUTES, 24 HOURS PER DAY.

Search

Main Menu

- Top Stories
- 24 Hours Overview
- Events Detection
- Most Active Themes
- Overview
- Advanced Search

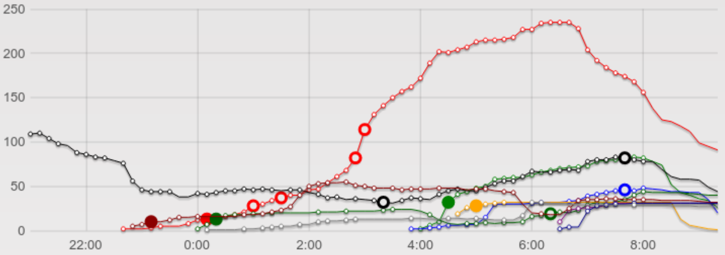
EU Focus

- EC President Ursula von der Leyen
- EC Commission Vice-President
- EC Commissioners
- EC News
- EP President
- Council President
- State of the Union Address
- Belgian Presidency of the Council of the EU

EU Policy Areas

- Antifraud (OLAF)
- Agriculture and Rural Development
- Better Regulation, Inter-

Current top 10 stories
Language: en Period: Mar 13, 2024, 2:20 AM - Mar 13, 2024, 2:20 PM



reset zoom

Putin Warns West: Russia Ready For Nuclear War, But No Rush

Articles : 340 | Last update : Mar 13, 2024, 1:07:00 PM | Start : Mar 13, 2024, 3:46:00 AM | Sources : 207 | Peak : 4 | Current rank : 1

President Putin says Russia is ready for a nuclear war
nbcnews Wednesday, March 13, 2024 at 1:07:00 PM Central European Standard Time | info [other]

Top terms

russia is ready | putin says russia | president vladimir putin | nuclear war | united states

More articles...

Palestinians killed in West Bank amid violence surge

Articles : 84 | Last update : Mar 13, 2024, 1:02:00 PM | Start : Mar 13, 2024, 9:00:00 AM | Sources : 73 | Peak : 1 | Current rank : 2

The Latest | Israeli forces kill 2 Palestinians during raid in occupied West Bank

Tools

Wednesday, March 13, 2024
2:33:00 PM Central European Standard Time

RSS | KML | MAP
Facebook

subscribe | manage

info

Languages

Select top stories in other languages.

ar bg cs da de el
en es et fi fr hr
hu it lt lv mt nl
pl pt ro ru sk sl
sv sw tr zh

Show additional languages

Interface:
en - English

Legend

Main Menu

- Top Stories
- 24 Hours Overview
- Events Detection
- Most Active Themes Overview
- Advanced Search

EU Focus

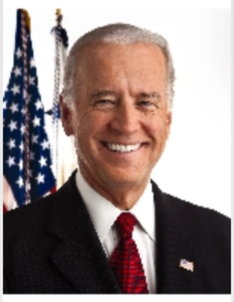
- EC President Ursula von der Leyen
- EC Commission Vice-President
- EC Commissioners
- EC News
- EP President
- Council President
- State of the Union Address
- Belgian Presidency of the Council of the EU

EU Policy Areas

- Antifraud (OLAF)
- Agriculture and Rural Development
- Better Regulation, Inter-Institutional Relations, the

Joseph Biden

Last updated on 2024-03-11T12:07+0100.



ABOUT THIS IMAGE
LICENSE: PUBLIC
AUTHOR: ANDREW CUTRARO, WHITE HOUSE PHOTOGRAPHER.

Extracted quotes from

Joe Biden said : "I believe that the American people will choose to keep us moving into the future,"
myjoyonline Wednesday, March 13, 2024 at 2:09:00 PM Central European Standard Time

Joe Biden asegurado (about Donald Trump) : "Me siento honrado de que la amplia coalición de votantes que representan la rica diversidad del Partido Demócrata en todo el país hayan depositado su fe en mí una vez más para liderar nuestro partido, y nuestro país , en un momento en el que la amenaza que supone el expresidente Donald Trump es mayor que nunca"
radioagricultura Wednesday, March 13, 2024 at 1:58:00 PM Central European Standard Time

Joe Biden déclaré (about Donald Trump) : «Je suis honoré que la large coalition de votants représentant la riche diversité du Parti démocrate à travers le pays aient placé leur foi en moi une fois encore pour conduire le

Key Titles and Phrases (Last 30) | Names (Top 30)

Key Titles and Phrases	Count	Lang	Last Seen
homólogo	0.00%	ES	11/03/2024
president	38.60%	EN	07/03/2024
presidente americano	0.00%	PT	03/03/2024
presidente	0.02%	ES	07/03/2024

Tools

Wednesday, March 13, 2024 2:35:00 PM Central European Standard Time

Facebook manage

Languages

Select your languages

ab ar az be bg bs ca cs da de el en eo es et fa fi fo fr ga gl ha he hi hr hu hy id is it ja ka km ko ku ky lb lo lt lv mk ml mn ms mt my na ne nl no ny os pap pl ps pt ro ru si sk sl so sq sr sv sw ta th tr tt uk ur vi zh zxx all

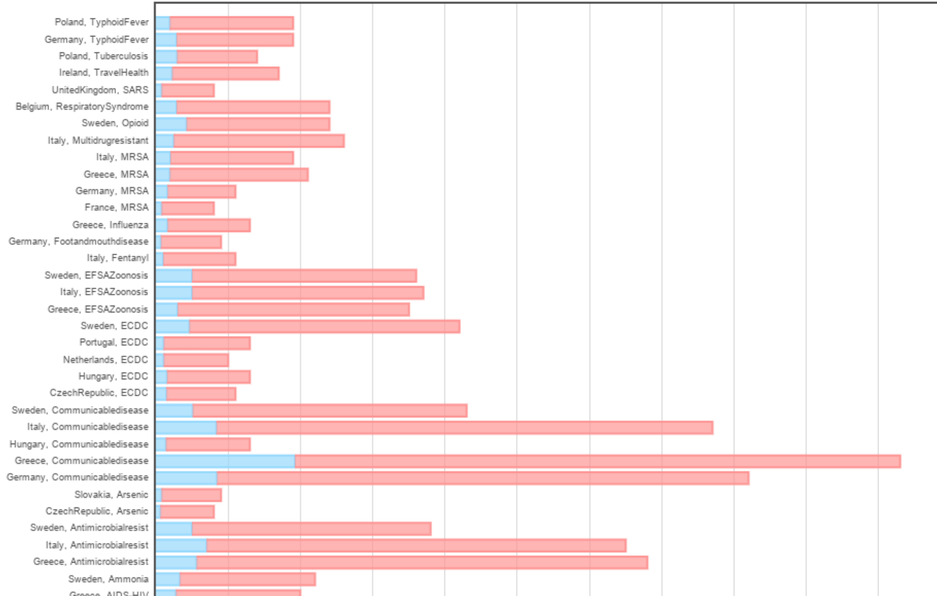
MEDISYS

Search

advanced search

- Top Stories
- Event Extraction
- Recent Disease Incidents
- Alert Statistics
- Communicable Diseases
- Symptoms
- Bioterrorism
- Nuclear
- Chemical
- ECDC
- EFSA
- EMCDDA
- ENV_RISKS
- Food Security
- SAM
- Medical Devices
- Vaccination
- Other
- Continents
- Official Sources
- Sources List

Today's Alert Statistics for European Union

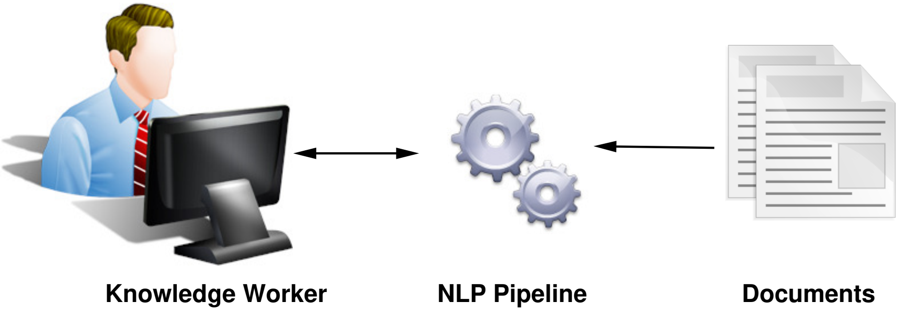


Recent Disease Incidents

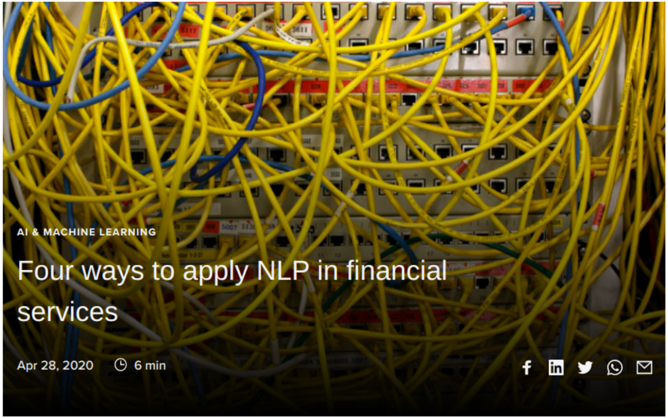
Automatically extracted from MediSys news items by PULS (University of Helsinki, Finland)

Disease	Time	Location	Status	Cases	Description
Dengue	--	Southeast Asia		five cases	Metro Doctors highlight dengue fever risk after five city cases Ruijin Hospital warns of risks and the need for timely and proper treatment after seeing five patients at its infectious disease department who had travelled to Southeast Asia. 4m
end-stage chronic kidney disease	--	Cyprus		more than 1,200 patients	Cyprus records more than 1,200 patients with end-stage chronic kidney disease
Rabies	2024.02	India	†	woman	Woman Dies Of Rabies 3 Days After Completing Vaccination Course; All About The Fatal Infection
Measles	2024.03.09	Ireland		passenger	UAE: Passenger tests positive for measles on Abu Dhabi-Dublin flight; alert issued
virus	2023.04.08	Bangladesh	†	one deaths	Bangladesh: One man takes charge of last rites for 87 virus victims
Coronavirus	2024.03.11	Iraq	†	eight people	Iraq extends entry ban on Iranians over coronavirus
mutated virus	2023.09.15-2024.03.13	Denmark		cases	Denmark: Cluster-5 coronavirus most likely eradicated

Architecture 2.0



→ Worksheet #8: Task 1



AI & MACHINE LEARNING

Four ways to apply NLP in financial services

Apr 28, 2020 · 6 min

Jo Stichbury

Freelance Technical Writer

Natural language processing (NLP) is increasingly used to review unstructured content or spot trends in markets. How is Refinitiv Labs applying NLP in financial services to meet challenges around investment decision-making and risk management?

So you want to build a Text Mining system. . .

René Witte

Requirements

A Natural Language Processing (NLP) system requires a large amount of infrastructure work:

- Document handling, in various formats (plain text, HTML, XML, PDF, . . .), from various sources (files, DBs, email, . . .)
- Annotation handling (stand-off markup)
- Component implementations for standard tasks, like Tokenizers, Sentence Splitters, Part-of-Speech (POS) Taggers, Finite-State Transducers, Full Parsers, Classifiers, Noun Phrase Chunkers, Lemmatizers, Entity Taggers, Coreference Resolution Engines, Summarizers, Deep Learning Models. . .

As well as *tools* and *resources* for concrete tasks and languages:

- Lexicons, WordNets
- Grammar files and Language models
- Machine Learning Algorithms
- Evaluation Metrics
- etc.

Existing Resources

Fortunately, you don't have to start from scratch

Many (open source) tools and resources are available:

NLP Tools: programs performing a single text processing task, like classifiers, parsers, or NP chunkers

NLP Libraries: collection of algorithms and resources for various tasks and languages

Frameworks: large-scale integration architectures for combining and controlling all components and resources of an NLP system

Resources: for various languages, like lexicons, wordnets, grammars, or pre-trained ML models

Cloud Computing Platforms: scalable, on-demand computing environments for data storage, processing, and analysis; capable of handling large datasets and computationally intensive NLP tasks

NLP Development

Major Frameworks

Two important frameworks are:

- GATE *(General Architecture for Text Engineering)*, under development since 1995 at University of Sheffield, UK
- UIMA *(Unstructured Information Management Architecture)*, developed by IBM; open-sourced in 2007 (Apache project)

Both frameworks are open source (GATE: LGPL, UIMA: Apache)

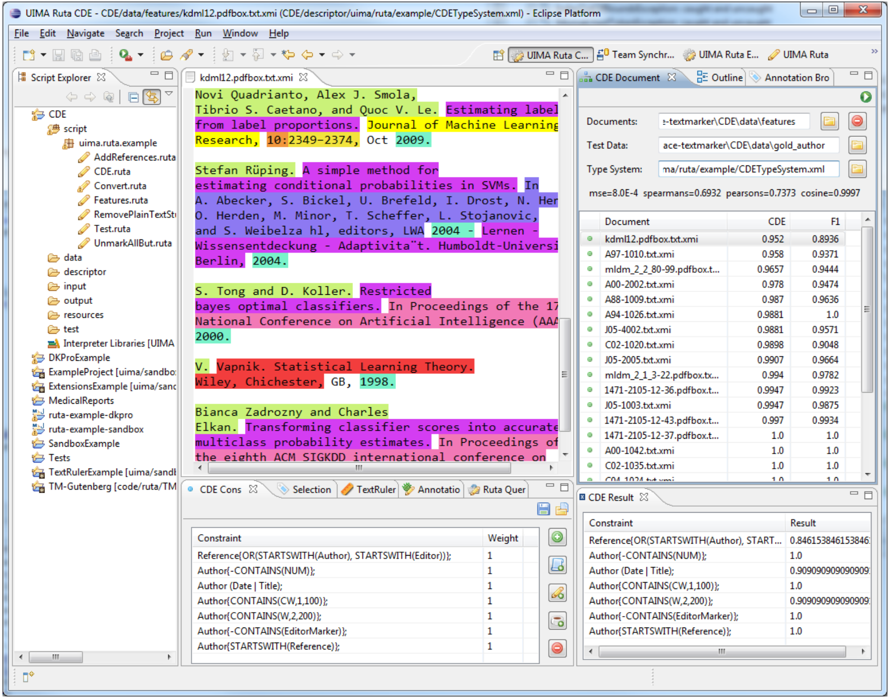
Libraries

- Numerous NLP libraries: NLTK (Python), Stanford CoreNLP, OpenNLP…
- Various integrations (e.g, CoreNLP has GATE wrapper, Python bindings)

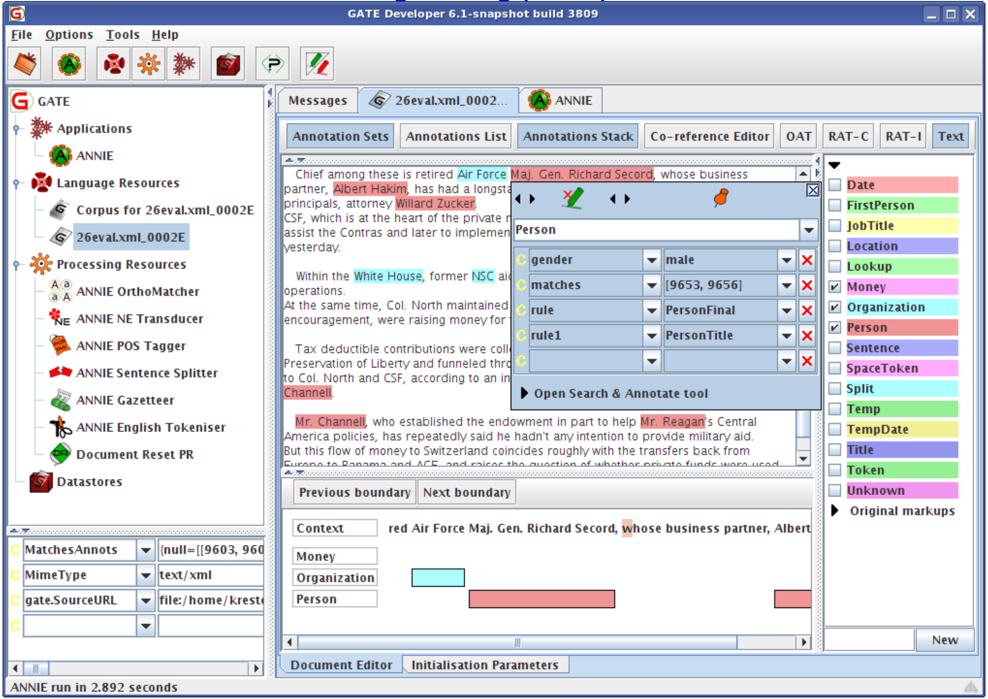
Current Trends

- Increasing use of Deep Learning for NLP (e.g, TensorFlow, PyTorch)
- Use of pre-trained language models (e.g., BERT), exchanged on HuggingFace
- Large Language Models (LLMs)
- Cloud Computing Solutions and Online APIs

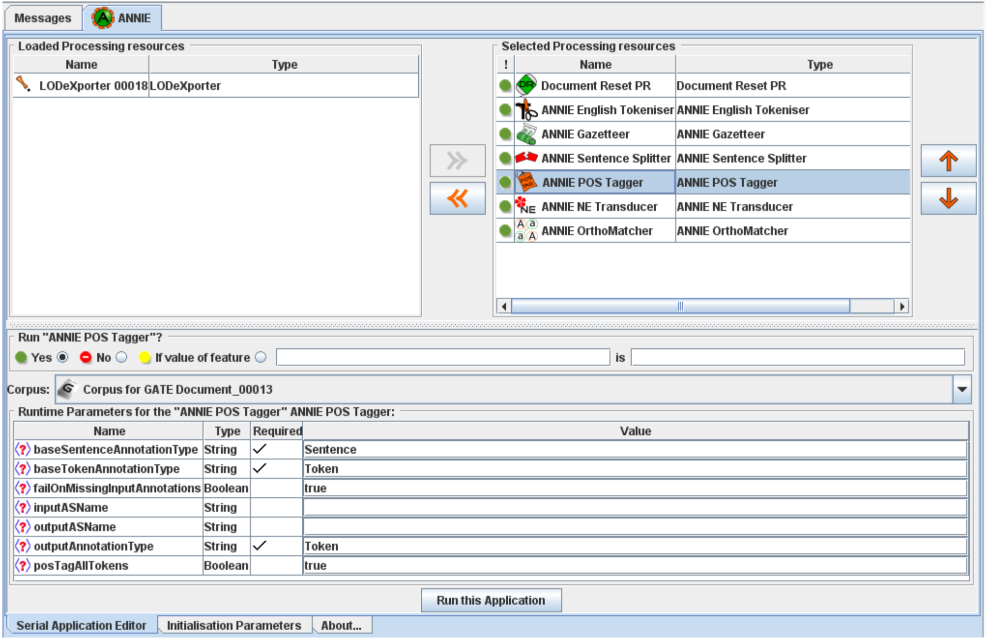
Unstructured Information Management Architecture (UIMA)



General Architecture for Text Engineering (GATE)



NLP Pipeline in GATE



Pipeline Step: Tokenization

René Witte

Université Concordia University

Example Tokenisation Rules

```
#numbers#
// a number is any combination of digits
"DECIMAL_DIGIT_NUMBER"+ >Token;kind=number;

#whitespace#
(SPACE_SEPARATOR) >SpaceToken;kind=space;
(CONTROL) >SpaceToken;kind=control;
```

Example Output

Type	Set	Start	End	Features
Token		158	163	{kind=word, length=5, orth=lowercase, string=years}
SpaceToken		163	164	{kind=space, length=1, string= }
Token		164	167	{kind=word, length=3, orth=lowercase, string=ago}
Token		167	168	{kind=punctuation, length=1, string=,}
SpaceToken		168	169	{kind=space, length=1, string= }
Token		169	180	{kind=word, length=11, orth=lowercase, string=researchers}
SpaceToken		180	181	{kind=space, length=1, string= }

1417 Annotations (0 selected)

Pipeline Step: POS Tagging

Producing POS Annotations

POS-Tagging assigns a part-of-speech-tag (POS tag) to each `Token`.

- GATE comes with the Hepple tagger for English, which is a modified version of the Brill tagger

Example output

Inc., the unit of New York-based Loews Corp. that makes Kent
cigarettes, stopped using crocidolite in its Micronite cigarette
filters in 1956.
Although preliminary findings were reported more than a year
ago, the latest results appear in today's New England Journal of

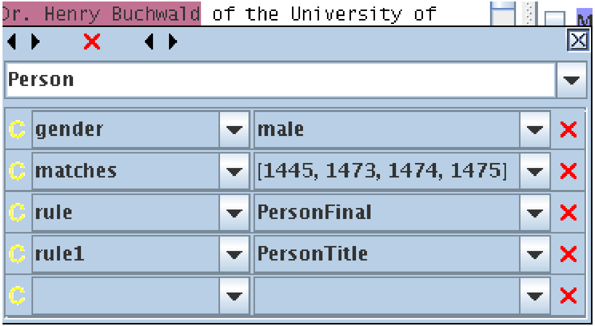
Type	Set	Start	End	Features
Token		485	494	{category=NN, kind=word, length=9, orth=upperInit
Token		495	504	{category=NN, kind=word, length=9, orth=lowercas
Token		505	512	{category=NNS, kind=word, length=7, orth=lowerca
Token		513	515	{category=IN, kind=word, length=2, orth=lowercase
Token		516	520	{category=CD, kind=number, length=4, string=1956
Token		520	521	{category=., kind=punctuation, length=1, string=.}

Pipeline Step: Named Entity (NE) Detection

Transducer-based NE Detection

Using all the information obtained in the previous steps (Tokens, Gazetteer lookups, POS tags), ANNIE now runs a sequence of JAPE-Transducers to detect Named Entities (NE)s.

Example for a detected *Person*



We can now look at the grammar rules that found this person.

Entity Detection: Finding Persons

Strategy

A JAPE grammar rule combines information obtained from POS-tags with Gazetteer lookup information

- although the last name in the example is not in any list, it can be found based on its POS tag and an additional first name/last name rule (not shown)
- many additional rules for other Person patterns, as well as Organizations, Dates, Addresses, ...

Persons with Titles

```
Rule:   PersonTitle
Priority: 35
(
 {Token.category == DT}|
 {Token.category == PRP}|
 {Token.category == RB}
)?
(
 (TITLE)+
 ((FIRSTNAME | FIRSTNAMEAMBIG
   | INITIALS2)
 )?
  (PREFIX)*
  (UPPER)
 (PERSONENDING)?
)
:person --> ...
```

→ Worksheet #8: Task 2

Outline

1. Introduction
2. **Natural Language Processing (NLP)**
   - Language Models
   - Tokenization
   - Sentence Splitting
   - Morphology
   - Part-of-Speech (POS) Tagging
   - Chunking and Parsing
   - Named Entity Recognition
   - Entity Linking
   - Pipelines
3. Text Mining Applications
4. Notes and Further Reading

NLP Applications

Search	Web	Documents	Autocomplete
Editing	Spelling	Grammar	Style
Dialog	Chatbot	Assistant	Scheduling
Writing	Index	Concordance	Table of contents
Email	Spam filter	Classification	Prioritization
Text mining	Summarization	Knowledge extraction	Medical diagnoses
Law	Legal inference	Precedent search	Subpoena classification
News	Event detection	Fact checking	Headline composition
Attribution	Plagiarism detection	Literary forensics	Style coaching
Sentiment analysis	Community morale monitoring	Product review triage	Customer care
Behavior prediction	Finance	Election forecasting	Marketing
Creative writing	Movie scripts	Poetry	Song lyrics

Industrial-Strength Natural Language Processing

IN PYTHON

Get things done

spaCy is designed to help you do real work — to build real products, or gather real insights. The library respects your time, and tries to avoid wasting it. It's easy to install, and its API is simple and productive.

GET STARTED

Blazing fast

spaCy excels at large-scale information extraction tasks. It's written from the ground up in carefully memory-managed Cython. If your application needs to process entire web dumps, spaCy is the library you want to be using.

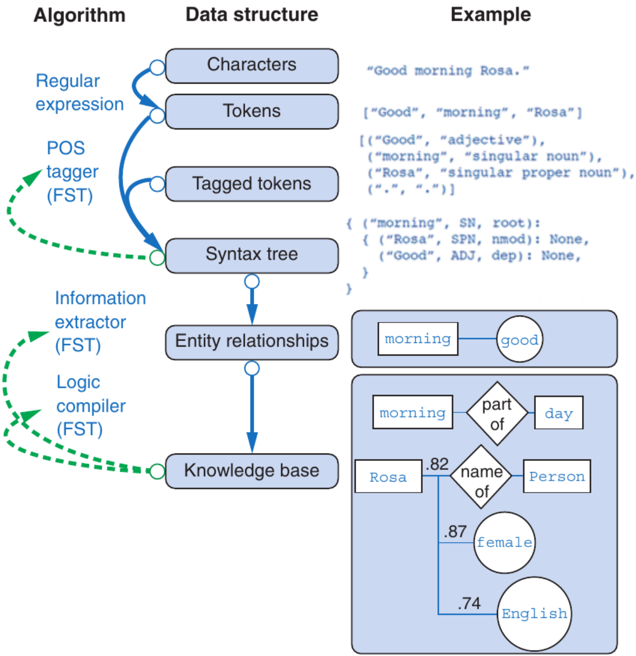
FACTS & FIGURES

Awesome ecosystem

Since its release in 2015, spaCy has become an industry standard with a huge ecosystem. Choose from a variety of plugins, integrate with your machine learning stack and build custom components and workflows.

READ MORE

Example NLP Pipeline



Applications

Cryptography, compression, spelling correction, predictive text, search, dialog (chatbot)

Search, stylistics, spam filter, sentiment analysis, word2vec math, semantic search, dialog (chatbot)

Spelling and grammar correction, stylistics, dialog (chatbot)

Question answering, stylistics, complex dialog, grammar correction, writing coach

Knowledge extraction and inference, medical diagnosis, question answering, game playing

Theorem proving, inference, natural language database queries, artificial general intelligence (AGI)

Language Models

Language dependent code

- Some parts of spaCy work language-independent
- But many steps require language-specific data, such as rules or pre-trained ML models

Need to load a language model to start, e.g., for English (small model):

```
import spacy
nlp = spacy.load("en_core_web_sm")
```

LANGUAGE	CODE	LANGUAGE DATA	PIPELINES
Chinese	zh	lang/zh	4 packages
Danish	da	lang/da	3 packages
Dutch	nl	lang/nl	3 packages
English	en	lang/en	4 packages
French	fr	lang/fr	4 packages
German	de	lang/de	4 packages
Greek	el	lang/el	3 packages
Italian	it	lang/it	3 packages
Japanese	ja	lang/ja	3 packages

Tokenization

Tokenization

Text is split into basic units called *Tokens*:

- word tokens
- number tokens
- space tokens
- ...

Consistent tokenization is important for all later processing steps

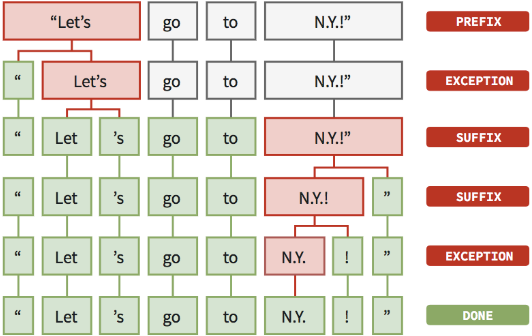
What is a word?

Unfortunately, even tokenization can be difficult:

- Is "John's" in *John's sick* one token or two?
  If one $\rightarrow$ problems in parsing (where's the verb?)
  If two $\rightarrow$ what do we do with *John's house?*
- What to do with hyphens?
  E.g., *database* vs. *data-base* vs. *data base*
- what to do with "C++", "A/C", ":-)", "..."?

Tokenization

```
doc = nlp(u"Let's go to N.Y.!")
print([token.text for token in doc])
```



→ **Worksheet #8: Task 3**

Sentence Splitting (a.k.a. Sentence Segmentation)

René Witte

Mark Sentence Boundaries

Detects sentence units. Easy case:

- often, sentences end with ".", "!", or "?"

Hard (or annoying) cases:

- difficult when a "." do not indicate an EOS:
  "MR. X", "3.14", "Y Corp.", . . .
- we can detect common abbreviations ("U.S."), but what if a sentence ends with one?
  ". . .announced today by the U.S. The. . .
- Sentences can be *nested* (e.g., within quotes)

Correct sentence boundary is important

for many downstream analysis tasks:

- POS-Taggers maximize probabilites of tags within a sentence
- Most Parsers work on individual sentences

See https://en.wikipedia.org/wiki/Sentence_boundary_disambiguation

Some difficult examples for sentence splitting

I live in the U.S. but I commute to work in Mexico on S.V. Australis for a woman from St. Bernard St. on the Gulf of Mexico.

I went to G.T.You?

She yelled "It's right here!" but I kept looking for a sentence boundary anyway.

I stared dumbfounded on as things like "How did I get here?," "Where am I?," "Am I alive?" flittered across the screen.

The author wrote "'I don't think it's conscious.' Turing said."

https://www.tm-town.com/natural-language-processing#golden_rules

→ Worksheet #8: Task 4

Morphological Analysis

René Witte

Morphological Variants

Words are changed through a morphological process called *inflection*:

- typically indicates changes in case, gender, number, tense, etc.
- example *car* $\rightarrow$ cars, *give* $\rightarrow$ *gives, gave, given*

Goal: “normalize” words

Stemming and Lemmatization

Two main approaches to normalization:

Stemming reduce words to a *base form*

Lemmatization reduce words to their *lemma*

Main difference: stemming just finds **any** base form, which doesn't even need to be a word in the language! Lemmatization find the actual *root* of a word, but requires morphological analysis.

Stemming vs. Lemmatization

René Witte

Concordia University

Stemming

Commonly used in Information Retrieval:

- Can be achieved with rule-based algorithms, usually based on suffix-stripping
- Standard algorithm for English: the *Porter* stemmer
- Advantages: simple & fast
- Disadvantages:
  - Rules are language-dependent
  - Can create words that do not exist in the language, e.g., *computers* → *comput*
  - Often reduces different words to the same stem, e.g.,
    army, arm → *arm*
    stocks, stockings → *stock*
- Stemming for other languages: *Lucene* and *Snowball* stemmer have rule files for many languages

Stemming vs. Lemmatization, Part II

René Witte

Lemmatization

Lemmatization is the process of deriving the base form, or *lemma*, of a word from one of its inflected forms. This requires a morphological analysis, which in turn typically requires a *lexicon*.

- Advantages:
  - identifies the *lemma* (root form), which is an actual word
  - less errors than in stemming
- Disadvantages:
  - more complex than stemming, slower
  - requires additional language-dependent resources
- While stemming is good enough for Information Retrieval, Text Mining often requires lemmatization
  - Semantics is more important (we need to distinguish an *army* and an *arm*!)
  - Errors in low-level components can multiply when running downstream

Morphology in spaCy

Morphological features are stored in the `MorphAnalysis` under `Token.morph`

Part-of-Speech (POS) Tagging

René Witte

Where are we now?

So far, we splitted texts into *tokens* and *sentences* and performed some *normalization*.

- Still a long way to go to an *understanding* of natural language...

Typical approach in text mining: deal with the complexity of language by applying intermediate processing steps to acquire more and more structure.

Next stop: *POS-Tagging*.

POS-Tagging

A statistical POS Tagger scans tokens and assigns POS Tags.
A black cat plays... → *A/DT black/JJ cat/NN plays/VB...*

- relies on different word order probabilities
- needs a manually tagged corpus for machine learning

Note: *this is not grammatical analysis (parsing)!*

Part-of-Speech (POS) Tagging (II)

René Witte

Tagsets

A tagset defines the tags to assign to words. Main POS classes are:

Noun	refers to entities like people, places, things or ideas
Adjective	describes the properties of nouns or pronouns
Verb	describes actions, activities and states
Adverb	describes a verb, an adjective or another adverb
Pronoun	word that can take the place of a noun
Determiner	describes the particular reference of a noun
Preposition	expresses spatial or time relationships

Note: real tagsets have from 45 (Penn Treebank) to 146 tags (C7).

POS Tagging Algorithms

René Witte

Concordia University

Fundamentals

POS-Tagging generally requires:

Training phase where a manually annotated corpus is processed by a machine learning algorithm; and a

Tagging algorithm that processes texts using learned parameters.

Performance is generally good (around 96%) when staying in the same domain.

Algorithms used in POS-Tagging

There is a multitude of approaches, commonly used are:

- Decision Trees
- Hidden Markov Models (HMMs)
- Support Vector Machines (SVM)
- Transformation-based Taggers (e.g., the Brill tagger)

POS Tagging in spaCy

```
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("Apple␣is␣looking␣at␣buying␣U.K.␣startup␣for␣$1␣billion")

for token in doc:
    print(token.text, token.lemma_, token.pos_, token.tag_, token.dep_,
            token.shape_, token.is_alpha, token.is_stop)
```

TEXT	LEMMA	POS	TAG	DEP	SHAPE	ALPHA	STOP
Apple	apple	PROPN	NNP	nsubj	Xxxxx	True	False
is	be	VERB	VBZ	aux	xx	True	True
looking	look	VERB	VBG	ROOT	xxxx	True	False
at	at	ADP	IN	prep	xx	True	True
buying	buy	VERB	VBG	pcomp	xxxx	True	False
U.K.	u.k.	PROPN	NNP	compound	X.X.	False	False
startup	startup	NOUN	NN	dobj	xxxx	True	False
for	for	ADP	IN	prep	xxx	True	True
$	$	SYM	$	quantmod	$	False	False

Tagsets

Understanding POS Tags

- There are different tagsets used by different tools
- spaCy has a built-in explanation method:

```
spacy.explain("NNP")
> noun, proper singular
```

- spaCy uses the Universal Dependency Scheme (https://universaldependencies.org/u/pos/)

→ Worksheet #8: Task 5

POS	DESCRIPTION	EXAMPLES
ADJ	adjective	big, old, green, incomprehensible, first
ADP	adposition	in, to, during
ADV	adverb	very, tomorrow, down, where, there
AUX	auxiliary	is, has (done), will (do), should (do)
CONJ	conjunction	and, or, but

Syntax: Chunking and Parsing

Finding Syntactic Structures

We can now start a syntactic analysis of a sentence using:

Parsing producing a *parse tree* for a sentence using a parser, a grammar, and a lexicon

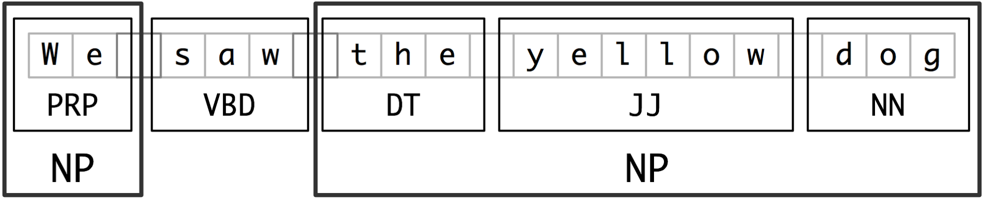
Chunking finding syntactic constituents like *Noun Phrases (NPs)* or *Verb Groups (VGs)* within a sentence

Chunking vs. Parsing

Chunking: Quick extraction of phrases, bypassing full parse trees

Full Parsing: Complete syntactic analysis, can fail due to inaccuracies or prior errors (grammatical inaccuracies, novel words, bad tokenization, wrong sentence splits, errors in POS tagging, . . .)

Partial Parsing: Balances depth and efficiency, less susceptible to errors



Noun Phrase Chunking

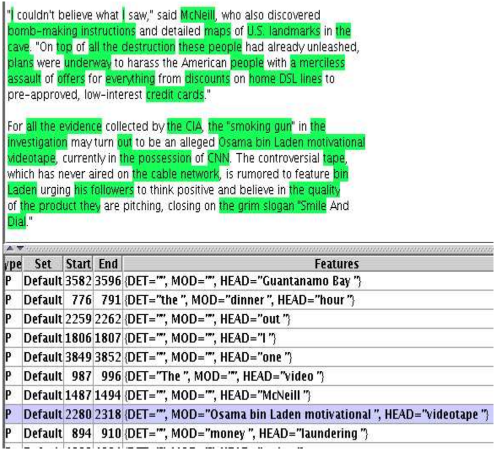
NP Chunker

Rule-based approach for finding NPs

Grammar Excerpt

```
(NP      (DET MOD HEAD))
(MOD     (MOD-ingredients)
         (MOD-ingredients MOD)
         ())
(HEAD    (NN) ...)
```

Example



NP Chunking in spaCy

René Witte
Concordia University


```python
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("Autonomous cars shift insurance liability toward manufacturers")
for chunk in doc.noun_chunks:
    print(chunk.text, chunk.root.text, chunk.root.dep_,
          chunk.root.head.text)
```

TEXT	ROOT.TEXT	ROOT.DEP_	ROOT.HEAD.TEXT
Autonomous cars	cars	nsubj	shift
insurance liability	liability	dobj	shift
manufacturers	manufacturers	pobj	toward

Chunking vs. Parsing, Round 2

René Witte

Concordia University

What can we do with chunks?

(NP) chunks are very useful in finding named entities (NEs), e.g., *Persons, Companies, Locations, Patents, Organisms, . . .*.

But additional methods are needed for finding relations:

- *Who* invented *X*?
- *What* company created product *Y* that is doomed to fail?
- *Which* organism is this protein coming from?

Parse trees can help in determining these relationships

Parsing Challenges

Parsing is hard due to many kinds of ambiguities:

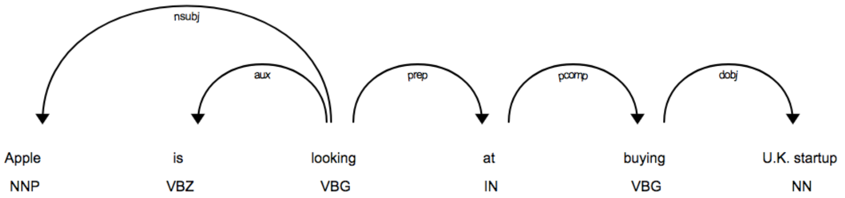
PP-Attachment which NP takes the PP? Compare:
He ate spaghetti with a fork.
He ate spaghetti with tomato sauce.

NP Bracketing *plastic cat food can cover*

Garden Paths *The old man the boat.*

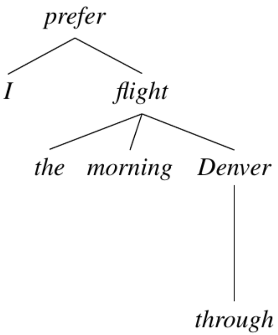
POS tags & dependencies

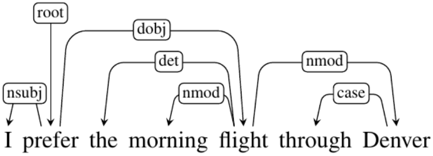
```
doc = nlp(u"Apple is looking at buying U.K. startup")

for token in doc:
    print(token.text, token.pos_, token.tag_)
```

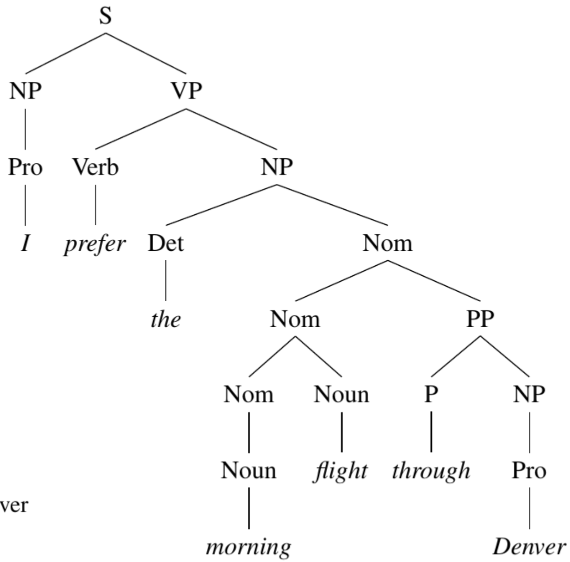


Dependency Parsing vs. Constituent-based Parse Tree

Parsing *"I prefer the morning flight through Denver."*







→ Worksheet #8: Task 6

Dependency vs. Constituent Parsing: Applications

Dependency Parsing

- Captures relationships between words
- Direct links denote syntactic structure
- Useful for understanding sentence dynamics

Constituent-based Parsing

- Organizes words into nested constituents
- Reflects phrase structure grammar
- Ideal for analyzing sentence composition

Why spaCy Uses Dependency Parsing

Efficiency: Faster, more scalable for processing large texts

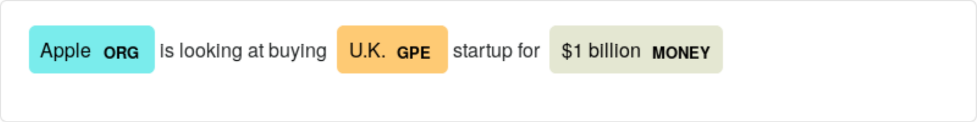
Universality: Adapts well across languages with different syntactic structures

Practicality: Directly useful for applications like information extraction and relation detection

Named Entity Recognition in spaCy

```python
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("Apple␣is␣looking␣at␣buying␣U.K.␣startup␣for␣$1␣billion")

for ent in doc.ents:
    print(ent.text, ent.start_char, ent.end_char, ent.label_)
```

TEXT	START	END	LABEL	DESCRIPTION
Apple	0	5	ORG	Companies, agencies, institutions.
U.K.	27	31	GPE	Geopolitical entity, i.e. countries, cities, states.
$1 billion	44	54	MONEY	Monetary values, including unit.



Applications: Enhancing search, content analysis, data extraction, . . .

Entity Models

Which entities are detected?

- Depends on model and training data
- spaCy's NER model is trained on OntoNotes-5.0, a diverse corpus with varied text genres (https://catalog.ldc.upenn.edu/LDC2013T19)

→ Worksheet #8: Task 7

TYPE	DESCRIPTION
PERSON	People, including fictional.
NORP	Nationalities or religious or political groups.
FAC	Buildings, airports, highways, bridges, etc.
ORG	Companies, agencies, institutions, etc.
GPE	Countries, cities, states.
LOC	Non-GPE locations, mountain ranges, bodies of water.
PRODUCT	Objects, vehicles, foods, etc. (Not services.)

To rule or not to rule...

How to find a new kind of Named Entity (NE)?

Two general solutions:

Rule-based: write rules (regular expressions, transducers) that capture as many variations as possible, with as few false positives as possible

Machine learning: train a machine learning model, using manually annotated examples (supervised learning)

Pros&Cons

- Rules can be developed quickly: good for proof-of-concept/demo, bootstrapping a ML corpus, easy (unambiguous) patterns
- Rules are "brittle", they do not generalize well
- ML solutions generally perform better (more robust with respect to variations)
- But a ML approach requires significant effort for creating training data, as well as some effort for feature engineering
- Hybrid approaches blend rule-based systems and machine learning to enhance accuracy, flexibility, and efficiency in NLP tasks

Developing Rules

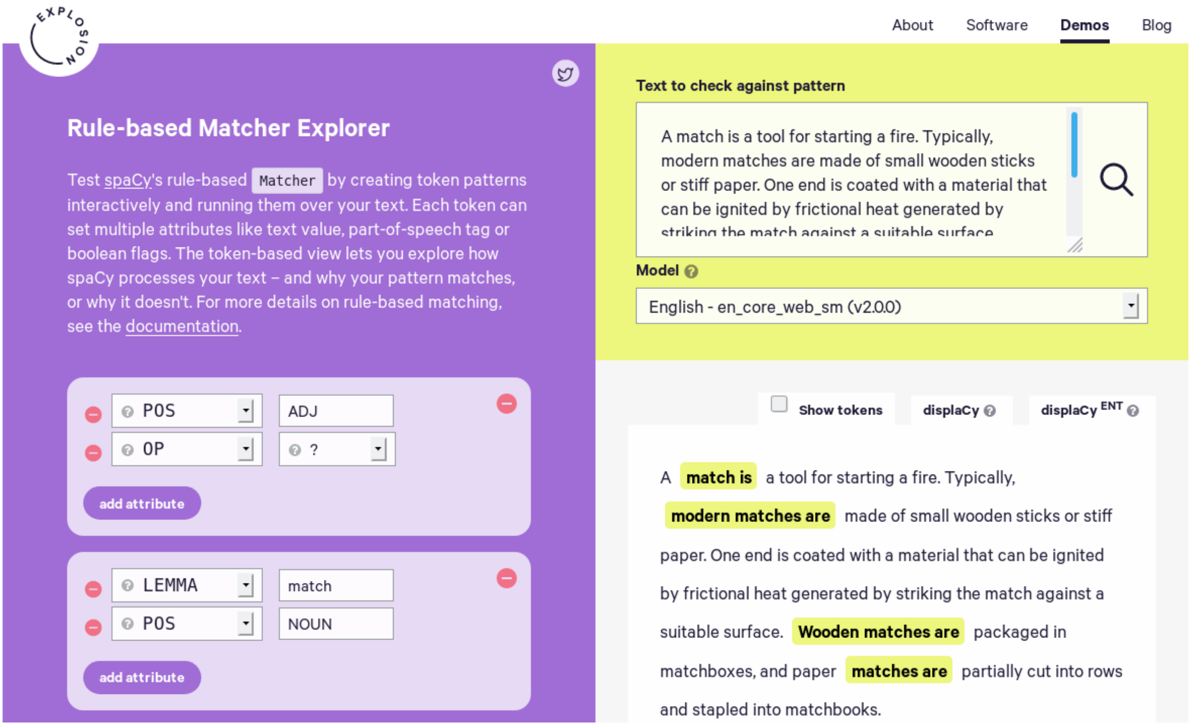
René Witte

Finite-state Transducers

In NLP, we generally use Finite-state Transducers (FSTs) for processing rules.

- Theory: Special kind of finite-state machine with input and output tape
- Practice: Unlike using regular expressions matching only the text, we match a graph, formed by the tokens, POS tags, dependency information, etc.

→ Worksheet #8: Task 8



About
Software
Demos
Blog
Rule-based Matcher Explorer
Test spaCy's rule-based Matcher by creating token patterns interactively and running them over your text. Each token can set multiple attributes like text value, part-of-speech tag or boolean flags. The token-based view lets you explore how spaCy processes your text – and why your pattern matches, or why it doesn't. For more details on rule-based matching, see the documentation.
POS
ADJ
OP
?
add attribute
LEMMA
match
POS
NOUN
add attribute
Text to check against pattern
A match is a tool for starting a fire. Typically, modern matches are made of small wooden sticks or stiff paper. One end is coated with a material that can be ignited by frictional heat generated by
Model
English - en_core_web_sm (v2.0.0)
Show tokens
displaCy
displaCy ENT
A match is a tool for starting a fire. Typically,
modern matches are made of small wooden sticks or stiff
paper. One end is coated with a material that can be ignited
by frictional heat generated by striking the match against a
suitable surface. Wooden matches are packaged in
matchboxes, and paper matches are partially cut into rows
and stapled into matchbooks.

Entity Linking

Grounding to a Knowledge Base

spaCy also provides an API for linking entities to a knowledge base, the `EntityLinker`

Example Pipeline

One implementation is https://pypi.org/project/spacy-entity-linker/

```
import spacy
from SpacyEntityLinker import EntityLinker
entityLinker = EntityLinker()
nlp = spacy.load("en_core_web_sm")
nlp.add_pipe(entityLinker, last=True, name="entityLinker")
doc = nlp("I␣watched␣the␣Pirates␣of␣the␣Carribean␣last␣silvester")

#returns all entities in the whole document
all_linked_entities=doc._.linkedEntities
for sent in doc.sents:
    sent._.linkedEntities.pretty_print()

#OUTPUT:
#https://www.wikidata.org/wiki/Q194318         194318
    Pirates of the Caribbean    Series of fantasy adventure films
#https://www.wikidata.org/wiki/Q12525597   12525597
    Silvester  the day celebrated on 31 December (Roman Catholic Church)
    or 2 January (Eastern Orthodox Churches)
```

spaCy fishing

Named entity disambiguation and linking on Wikidata in spaCy with Entity-Fishing.

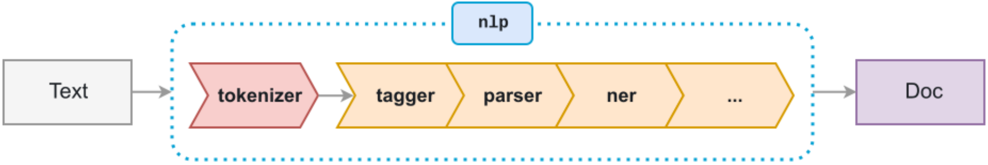
release v0.1.8 license MIT Stars 142

A spaCy wrapper of Entity-Fishing for named entity disambiguation and linking against a Wikidata knowledge base.

EXAMPLE

```
import spacy
text = 'Victor Hugo and Honoré de Balzac are French writers who lived in Paris.'
nlp = spacy.load('en_core_web_sm')
nlp.add_pipe('entityfishing')
doc = nlp(text)
for span in doc.ents:
    print((ent.text, ent.label_, ent._.kb_qid, ent._.url_wikidata, ent._.nerd_score))
# ('Victor Hugo', 'PERSON', 'Q535', 'https://www.wikidata.org/wiki/Q535', 0.972)
# ('Honoré de Balzac', 'PERSON', 'Q9711', 'https://www.wikidata.org/wiki/Q9711', 0.9724)
# ('French', 'NORP', 'Q121842', 'https://www.wikidata.org/wiki/Q121842', 0.3739)
# ('Paris', 'GPE', 'Q90', 'https://www.wikidata.org/wiki/Q90', 0.5652)
## Set parameter `extra_info` to `True` and check also span._.description, span._.src_description, span._.normal_term, span
```

Pipelines in spaCy



NAME	COMPONENT	CREATES	DESCRIPTION
tokenizer	`Tokenizer`	`Doc`	Segment text into tokens.
tagger	`Tagger`	`Doc[i].tag`	Assign part-of-speech tags.
parser	`DependencyParser`	`Doc[i].head`, `Doc[i].dep`, `Doc.sents`, `Doc.noun_chunks`	Assign dependency labels.
ner	`EntityRecognizer`	`Doc.ents`, `Doc[i].ent_iob`, `Doc[i].ent_type`	Detect and label named entities.
textcat	`TextCategorizer`	`Doc.cats`	Assign document labels.
...	custom components	`Doc._.xxx`, `Token._.xxx`, `Span._.xxx`	Assign custom attributes, methods or properties.

Pipelines in spaCy (contd.)

Working with pipelines

Loading a model defines the pipeline to be used in its metadata:

```
"pipeline": ["tagger", "parser", "ner"]
```

Processing a text will then apply each component in the pipeline in turn:

```
doc = nlp.make_doc("This␣is␣a␣sentence")
for name, proc in nlp.pipeline:
    doc = proc(doc)
```

You can disable components you don't need:

```
nlp = spacy.load("en_core_web_sm", disable=["parser"])
```

And of course add your own components (here at the end):

```
nlp.add_pipe(my_component, name="My␣new␣component", last=True)
```

More on pipelines: https://spacy.io/usage/processing-pipelines

Writing a spaCy component

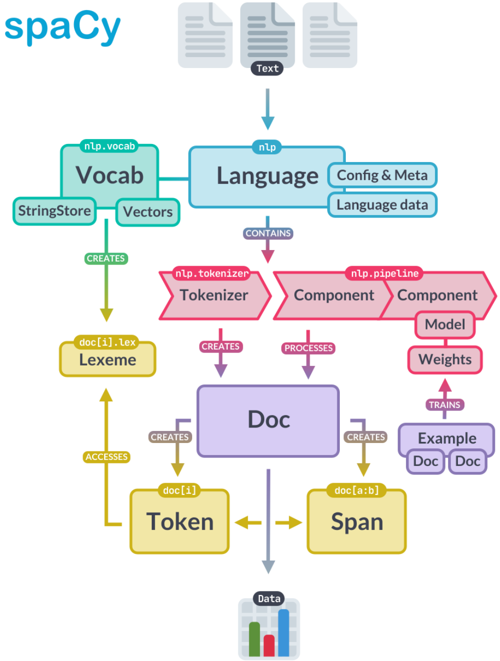
A simple "info" component

```python
import spacy
from spacy.language import Language

@Language.component("info_component")
def my_component(doc):
    print(f"After tokenization, this doc has {len(doc)} tokens.")
    print("The part-of-speech tags are:", [token.pos_ for token in doc])
    if len(doc) < 10:
        print("This is a pretty short document.")
    return doc

nlp = spacy.load("en_core_web_sm")
nlp.add_pipe("info_component", name="print_info", last=True)
print(nlp.pipe_names)  # ['tagger', 'parser', 'ner', 'print_info']
doc = nlp("This is a sentence.")
```

Output

```
['tok2vec', 'tagger', 'parser', 'attribute_ruler', 'lemmatizer',
 'ner', 'print_info']
After tokenization, this doc has 5 tokens.
The part-of-speech tags are: ['PRON', 'AUX', 'DET', 'NOUN', 'PUNCT']
This is a pretty short document.
```

Summary: spaCy Architecture



Outline

1. Introduction
2. Natural Language Processing (NLP)
3. **Text Mining Applications**
   - Example: Scientific Literature Mining
   - Mining Health Documents
   - Summary
4. Notes and Further Reading

Text Mining Biomedical Literature for Protein Mutations

Excerpts from PubMed journal PMID:14592457

> ...glutathione S-transferase (GST) fusion proteins in Escherichia coli and purified by GSH–agarose affinity chromatography. Mutant Q15K-W37R and mutant Q15R-W37R showed comparable activity for NAD and NADP with an increase in activity nearly 3fold over that of the wild type.

(Orange: Mutation, Red: Enzyme, Blue: Organism, Violet: Impact expression, Purple: Protein property, Green: Physical quantity)

What we are looking for?

Impact	Mutant Q15K-W37R and mutant Q15R-W37R showed... an increase in activity 3fold over that of the wild type.
Organism	Escherichia coli
Mutation	Q15K/W37R,Q15R/W37R
Enzyme	glutathione S-transferase (GST)
Protein property	activity
Physical Quantity	3fold
Impact Expression	increase

Enzymatic mechanism of low-activity mouse alcohol dehydrogenase 2

Stroemberg, P.; Svensson, S.; Berst, K.B.; Plapp, B.V.; Höög, J.O.; *Biochemistry* 43, 1323-1328 (2004)

Data extracted from this reference:

Engineering

Amino acid exchange	Commentary	Organism
P47A	site-directed mutagenesis, about 100fold increased activity compared to the wild-type enzyme	Mus musculus
P47H	site-directed mutagenesis, about 100fold increased activity compared to the wild-type enzyme	Mus musculus
P47Q	site-directed mutagenesis, about 100fold increased activity compared to the wild-type enzyme	Mus musculus

Inhibitors

Inhibitors	Commentary	Organism	Structure
cyclohexylformamide	dead-end inhibition pattern	Mus musculus	✓
Octanoic acid	dead-end inhibition pattern	Mus musculus	✓

Metals/Ions

Metals/Ions	Commentary	Organism	Structure
Zn2+	catalytic zinc ion	Mus musculus	✓

Organism

Organism	Primary Accession No. (UniProt)	Commentary	Textmining
Mus musculus	-	low-activity isozyme ADH2	-

Organism Entity Detection

René Witte

Université Concordia University

Organism Examples

$$\underbrace{\overbrace{\text{Emericella}}^{\text{genus}}\;\overbrace{\text{(Aspergillus)}}^{\text{old genus name}}\;\overbrace{\text{nidulans}}^{\text{species}}}_{\text{organism mention}}$$

$$\underbrace{\overbrace{\text{Escherichia}}^{\text{genus}}\;\overbrace{\text{coli}}^{\text{species}}\;\overbrace{\text{XLI-Blue}}^{\text{strain}}}_{\text{organism mention}}$$

Finding Organisms: Rule matching

Priority	Pattern
5	(GENUS) (SPECIES) (SUBSPECIES) (STRAIN)?
4	(GENUS) ("(")(GENUS)(")") (SPECIES) (STRAINKEYWORD)? (STRAIN) (STRAINKEYWORD)?
3	(SPECIES) (STRAINKEYWORD) (STRAIN)
2	(GENUS) (STRAINKEYWORD)? (STRAIN)
1	(FULLNAME) (STRAINKEYWORD)? (STRAIN) (STRAINKEYWORD)?

Query Interface (https://www.semanticsoftware.info/omm-query)

“Show me all protein mutations that impacted the protein property *affinity*”

Searching index: PMC-2012-06-18

{Impact} OVER affinity

Search

Results 1 - 10 of 6,993

2796128 (cached)

PIKKs (27). **The caffeine resistance mutations in TOR1 decrease affinity for caffeine or confer increased TORC1 kinase activity.** The W2176R mutation

2796128 (cached)

TORC1 kinase activity. **The W2176R mutation in TOR1 decreased affinity for caffeine.** Trp2176 is conserved in

2855616 (cached)

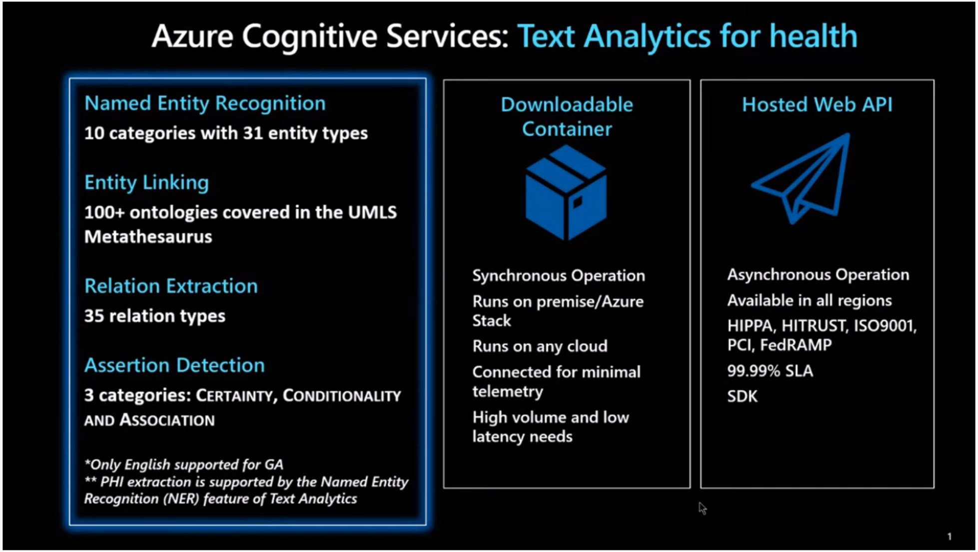
reported (7). **The ParM variant, labeled with two tetramethylrhodamines, binds ADP with relatively weak affinity (dissociation constant 30 μM) but responds to ADP binding with ~15-fold signal increase.** This means that the tetramethylrhodamine

2855616 (cached)

variant for tetramethylrhodamine labeling. **The same mutations had been successful in decreasing ATP affinity in the MDCC-ParM biosensor (7).** The new ParM mutant

[Naderi, Nona (2011) *Automated Extraction of Protein Mutation Impacts from the Biomedical Literature.* Masters thesis, Concordia University. https://spectrum.library.concordia.ca/35931/]

Text Analytics for Health (Microsoft Developer)



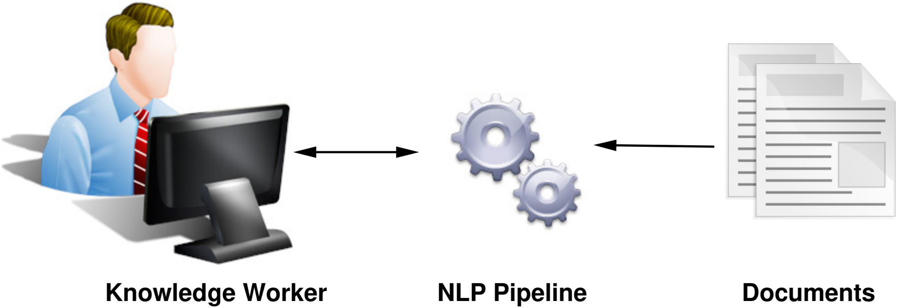
#MSBuild
What's new in Text Analytics for Health

https://www.youtube.com/watch?v=_RIBf01nxI0

Summary

Building a text mining system

- We have mature tools&resources to build robust, scalable systems now
- Prototyping a basic system demo can be done “in hours”
- Of course, some tasks are still complex R&D problems
- Cloud APIs are another option (convenient, but added cost and confidentiality concerns)
- Many businesses are still not aware of the potentials in analyzing their documents (automation, knowledge discovery)



René Witte
Concordia University

Outline

1. Introduction
2. Natural Language Processing (NLP)
3. Text Mining Applications
4. **Notes and Further Reading**

Reading Material

Required

- [LHH19, Chapter 11] (Information extraction)

Supplemental

- [PS12, Chapter 6] (Annotation and Adjudication)
- [JM, Chapter 14] (Dependency Parsing)

References

[JM] Daniel Jurafsky and James H. Martin.
Speech and Language Processing.
Third Edition draft, Jan 12, 2022.
https://web.stanford.edu/~jurafsky/slp3/.

[LHH19] Hobson Lane, Cole Howard, and Hannes Max Hapke.
Natural Language Processing in Action.
Manning Publications Co., 2019.
https://concordiauniversity.on.worldcat.org/oclc/1102387045.

[PS12] James Pustejovsky and Amber Stubbs.
Natural Language Annotation for Machine Learning.
O'Reilly, 2012.
https://concordiauniversity.on.worldcat.org/oclc/801812987.