Lecture 9

Text Mining

COMP 474/6741, Winter 2022

René Witte
Department of Computer Science
and Software Engineering
Concordia University

Outline

René Witte

Slides Credit

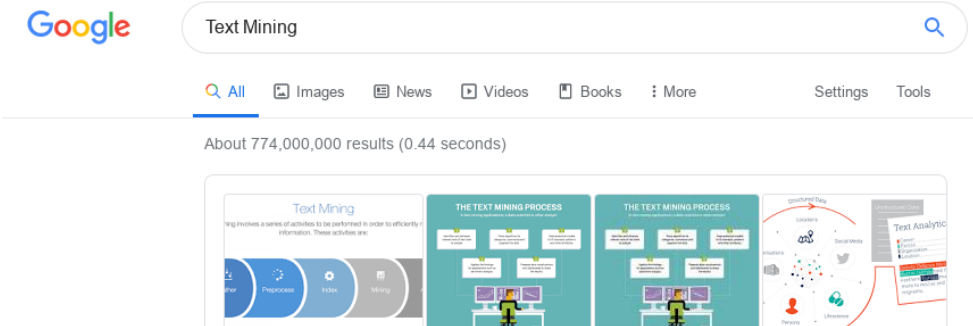
- Includes slides from Hoifung Poon, Chris Quirk & Scott Wen-Tau Yih, *Machine Reading for Precision Medicine*, https://www.microsoft.com/en-us/research/uploads/prod/2018/01/1802_aaai-tutorial_precision-med.pdf
- Includes slides from Matthew Honnibal & Ines Montani, *An introduction to spaCy*, https://github.com/explosion/talks/blob/master/2017-08-28_spaCy-101.p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

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*)



René Witte

Architecture 1.0

René Witte

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.

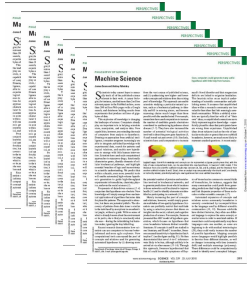
Gefitinib can treat tumors w. EGFR-L858E mutation

PubMed

27 million abstracts

Two new abstracts every minute

Adds over one million every year



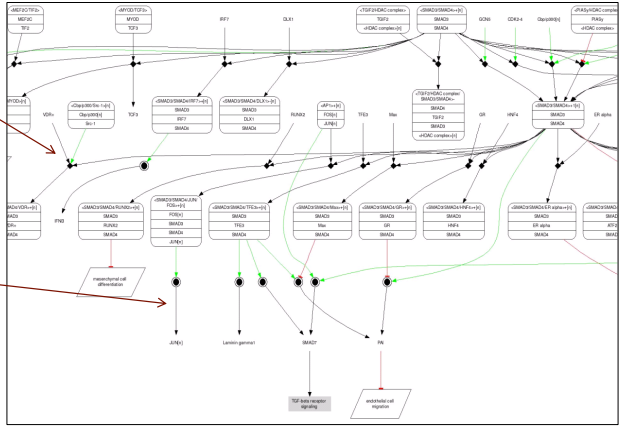
Machine Reading

PMID: 123

…
VDR+ binds to SMAD3 to form
…

PMID: 456

…
JUN expression is induced by SMAD3/4
…



Knowledge Base

Main Menu

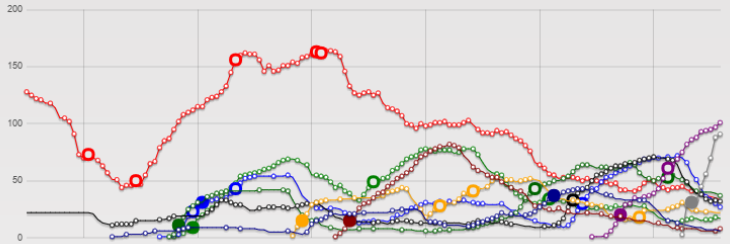
Top Stories
24 Hours Overview
Events Detection
Most Active Themes
Help about EMM
Overview
Advanced Search
Sources list
Web Site Map

EU Focus

EU Policy Areas

Agriculture and Rural Development
Better Regulation, Inter-Institutional Relations, the Rule of Law and the Charter of Fundamental Rights
Budget and Human Resources
Climate Action and Energy
Competition
Digital Economy and Society
Economic and Financial Affairs, Taxation and Customs
Education, Culture, Youth and Sport
Environment, Maritime Affairs and Fisheries
EU Foreign Affairs and Security Policy
DG Neighbourhood and Enlargement Negotiations
Financial Stability, Financial

Biggest 10 Stories Over Last 24h
Language: en Period: Nov 6, 2018 12:20 AM - Nov 7, 2018 12:20 AM



Oil prices drop over one percent on Iran sanctions waivers
Articles : 1527 | Last update : Nov 6, 2018 11:22:00 PM | Start : Nov 4, 2018 10:53:00 PM | Sources : 352

Oil market cautious as US punitive sanctions on Iran become effective
mercopress Tuesday, November 6, 2018 10:03:00 PM CET | info [other]

Oil prices were mixed on Monday after a steep five-day fall, as the United States formally imposed punitive sanctions on Iran but granted eight countries temporary waivers allowing them to keep buying oil from the Islamic Republic. The sanctions are part of U.S....

More articles...

Facebook blocks 115 accounts ahead of US midterm elections
Articles : 211 | Last update : Nov 7, 2018 12:18:00 AM | Start : Nov 6, 2018 5:12:00 AM | Sources : 165

Facebook blocks 115 accounts ahead of US midterm elections
ohio Tuesday, November 6, 2018 9:00:00 PM CET | info [other]

Facebook, Twitter and other companies have been fighting misinformation and election meddling on their services for the past two years. LONDON Facebook said it blocked 115 accounts for suspected "coordinated inauthentic behavior" linked to foreign groups attempting to interfere in Tuesday's U.S. midterm elections....

Tools

Wednesday, November 7, 2018 12:48:00 AM CET

RSS | MAP
Facebook
manage
info

Available on the App Store
ANDROID APP ON Google play

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

Country Watch

The country most in the news at the moment.

Main Menu

- Top Stories
- 24 Hours Overview
- Events Detection
- Most Active Themes
- Help about EMM
- Overview
- Advanced Search
- Sources list
- Web Site Map

EU Focus

- President Ursula von der Leyen
- Commission Vice-Presidents
- Commissioners
- EC News
- EP Presidency - David Sassoli
- Council President
- Portuguese Presidency of the Council of the EU

EU Policy Areas

- Agriculture and Rural Development

Anthony Fauci

Last updated on 2019-10-29T05:06+0100.

ABOUT THIS IMAGE
LICENSE: PUBLIC
AUTHOR: WHITE HOUSE PHOTO BY SHEALAH CRAIGHEAD

Extracted quotes from
twinCities Monday, March 22, 2021 7:25:00 PM CET

Anthony Fauci said : "There are very many countries in Europe and throughout the world who have already authorized this, so the fact that a United States-run study has confirmed the efficacy and the safety of this vaccine I think is an important contribution to global health in general,"
star-telegram Monday, March 22, 2021 7:17:00 PM CET

Anthony Fauci said : "The FDA will put a great deal of scrutiny in every aspect of this data,"
koaa Monday, March 22, 2021 5:42:00 PM CET

Anthony Fauci said (about **Rand Paul**) : "Sen. Paul has this message that we don't need masks, which goes against just about everything we know about how to prevent spread of the virus," "He was saying if you've been infected, or you've been vaccinated, don't wear a mask — which is completely against all public health tenets"
koaa Monday, March 22, 2021 5:42:00 PM CET

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

NAMES	LANG	COUNT
Anthony Fauci	EN	60.64%
Anthony Fauci	ES	11.75%
Anthony Fauci	FR	8.08%

Tools

Tuesday, March 23, 2021 5:30:00 PM CET

Facebook

manage

Available on the App Store | ANDROID APP ON Google play

Languages

Select your languages

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

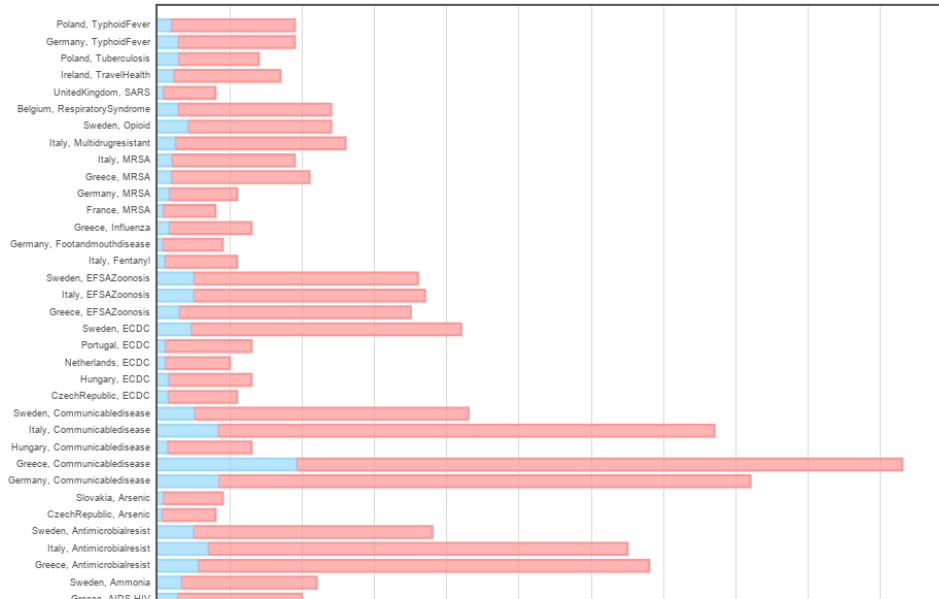
MEDISYS

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

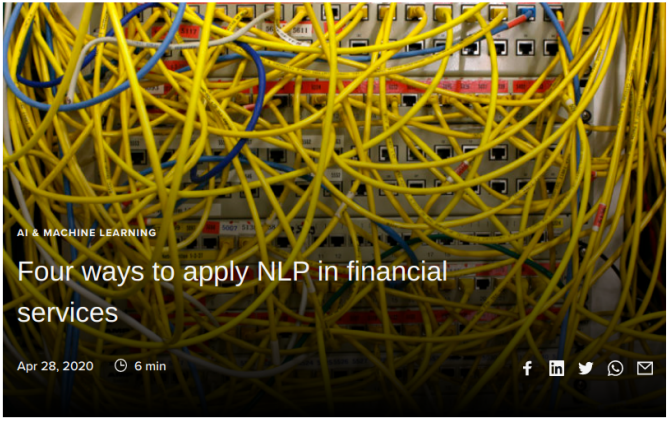


Architecture 2.0

Knowledge Worker **NLP Pipeline** **Documents**

→ Worksheet #8: Task 1

René Witte
Concordia University



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?

Solutions

Refinitiv Labs

Refinitiv™ Labs collaborate with customers around the world to solve big problems and rapidly prototype and validate solutions using data science and lean techniques

René Witte
Concordia University

Requirements

An 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, . . .

As well as *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 task, like classifiers, parsers, or NP chunkers

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

Frameworks: 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

René Witte
Concordia University

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)

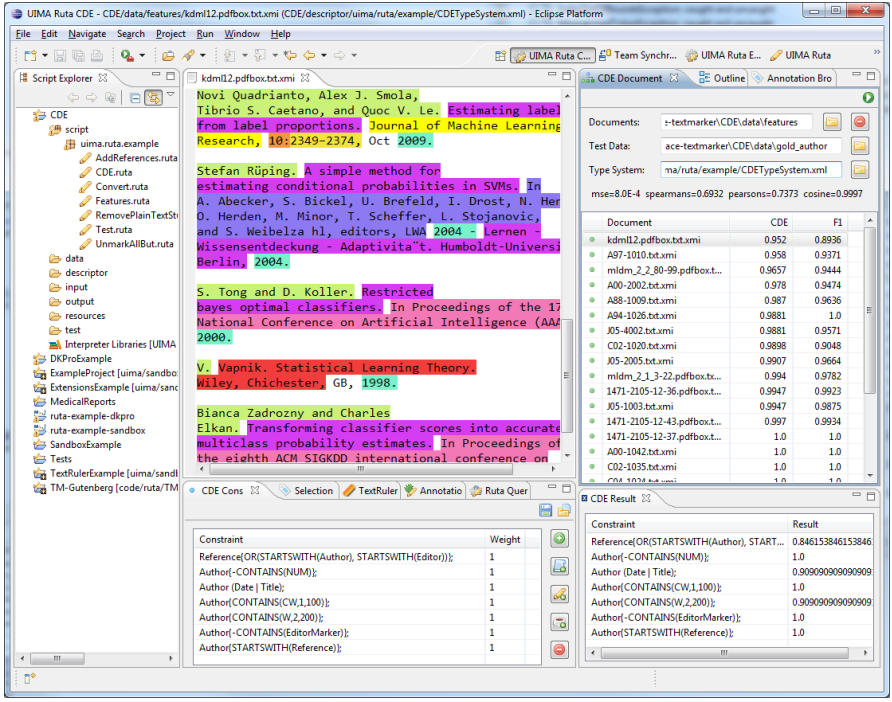
Libraries

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

Current Trends

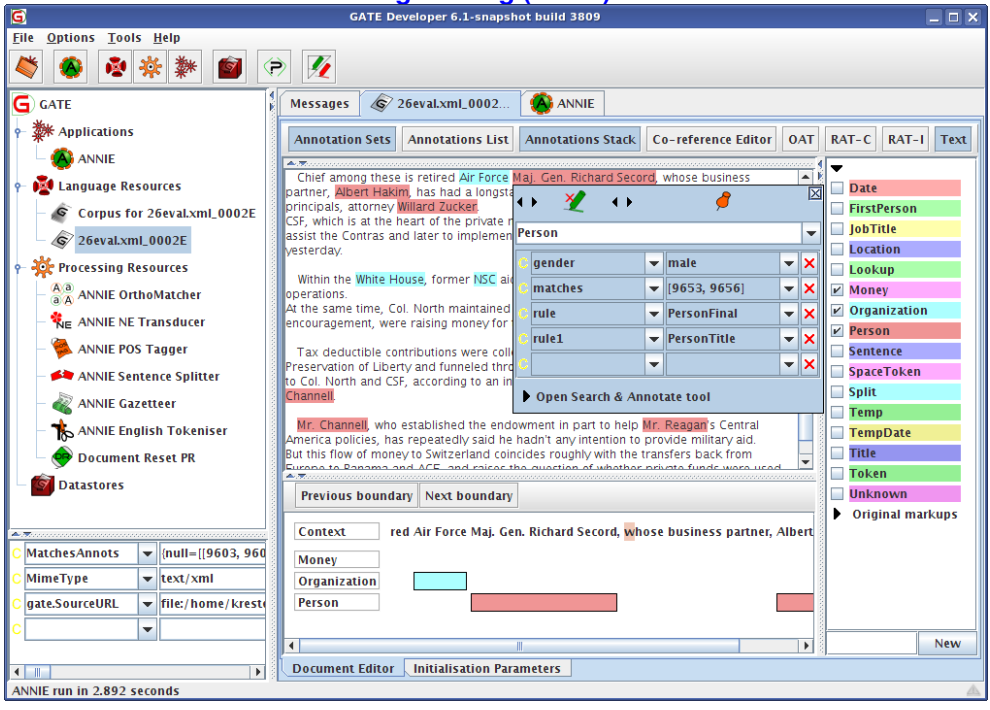
- Increasing use of Deep Learning tools/frameworks for NLP
- Keras/TensorFlow, PyTorch etc.

Unstructured Information Management Architecture (UIMA)



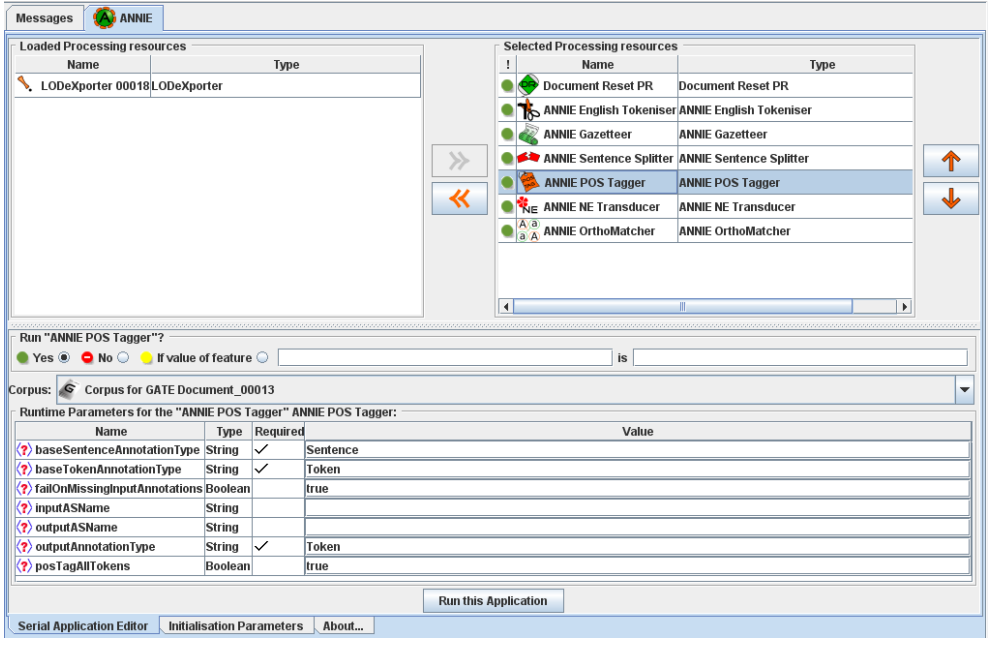
René Witte

General Architecture for Text Engineering (GATE)



René Witte

NLP Pipeline in GATE



René Witte

Concordia University

Pipeline Step: Tokenization

René Witte
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

René Witte

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=.}
Token		523	531	{category=IN, kind=word, length=8, orth=upperIniti

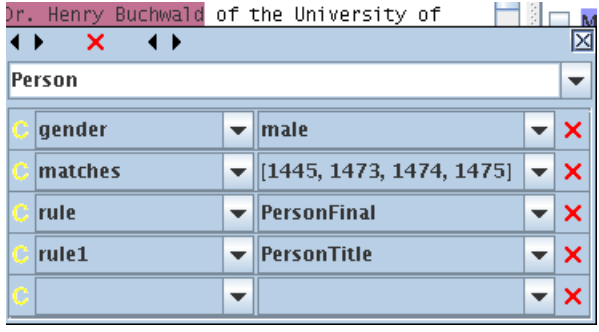
Pipeline Step: Named Entity (NE) Detection

René Witte

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

René Witte

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

René Witte

Concordia University

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

René Witte

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.

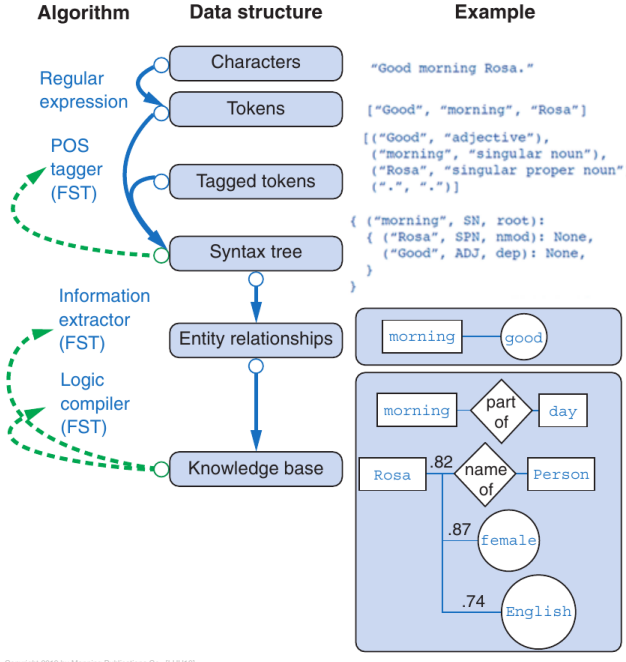
FACTS & FIGURES

Awesome ecosystem

In the five years since its release, 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)

René Witte

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:

```
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

René Witte
Concordia University

René Witte
Concordia University

Tokenization

Text is split into basic units called *Tokens*:

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

Consistent tokenization is important for all later processing steps

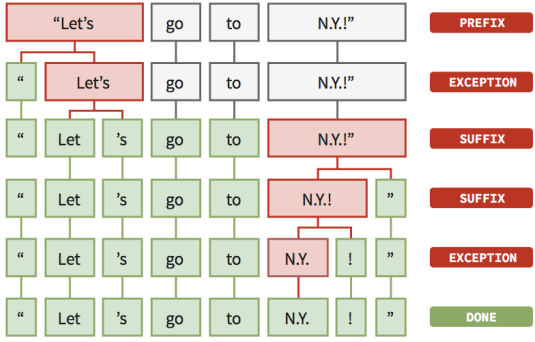
What is a word?

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**

René Witte

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

René Witte
Concordia University

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

René Witte
Concordia University

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**

René Witte
Concordia University

Morphological Variants

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

- typically indicates changes in case, gender, number, tense, etc.
- example *car* → cars, *give* → *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

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* $\rightarrow$ *comput*
  - Often reduces different words to the same stem, e.g.,
    army, arm $\rightarrow$ *arm*
    stocks, stockings $\rightarrow$ *stock*
- Stemming for other languages: *Lucene* and *Snowball* stemmer have rule files for many languages

Stemming vs. Lemmatization, Part II

René Witte
Concordia University

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

René Witte
Concordia University

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... $\rightarrow$ *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 parsing!*

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

René Witte
Concordia University

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

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

René Witte

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

René Witte

René Witte

Finding Syntactic Structures

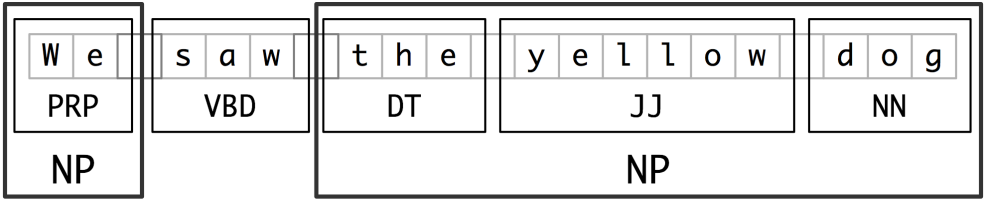
We can now start a syntactic analysis of a sentence using:

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

Producing a *full parse tree* often fails due to grammatical inaccuracies, novel words, bad tokenization, wrong sentence splits, errors in POS tagging, . . .
Hence, *chunking* and *partial parsing* are more commonly used.



Noun Phrase Chunking

René Witte
Concordia University

NP Chunker

Rule-based approach for finding NPs

Grammar Excerpt

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

Example

"I couldn't believe what I saw," said McNeill, who also discovered bomb-making instructions and detailed maps of U.S. landmarks in the cave. "On top of all the destruction these people had already unleashed, plans were underway to harass the American people with a merciless assault of offers for everything from discounts on home DSL lines to pre-approved, low-interest credit cards."

For all the evidence collected by the CIA, the "smoking gun" in the investigation may turn out to be an alleged Osama bin Laden motivational videotape, currently in the possession of CNN. The controversial tape, which has never aired on the cable network, is rumored to feature bin Laden urging his followers to think positive and believe in the quality of the product they are pitching, closing on the grim slogan "Smile And Dial."

ype	Set	Start	End	Features
P	Default	3582	3596	{DET="", MOD="", HEAD="Guantanamo Bay "}
P	Default	776	791	{DET="the ", MOD="dinner ", HEAD="hour "}
P	Default	2259	2262	{DET="", MOD="", HEAD="out "}
P	Default	1806	1807	{DET="", MOD="", HEAD="I "}
P	Default	3849	3852	{DET="", MOD="", HEAD="one "}
P	Default	987	996	{DET="The ", MOD="", HEAD="video "}
P	Default	1487	1494	{DET="", MOD="", HEAD="McNeill "}
P	Default	2280	2318	{DET="", MOD="Osama bin Laden motivational ", HEAD="videotape "}
P	Default	894	910	{DET="", MOD="money ", HEAD="laundering "}

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:

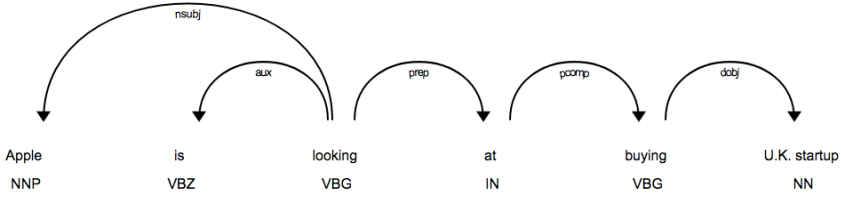
PP-Attachement which NP takes the PP? Compare:

He ate spaghetti with a fork.
He ate spaghetti with tomato sauce.

NP Bracketing *plastic cat food can cover*

POS tags & dependencies

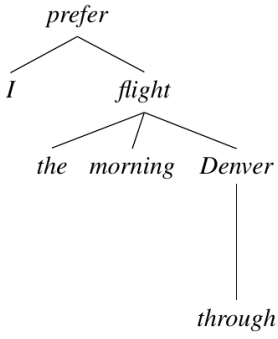
```
doc = nlp(u"Apple is looking at buying U.K. startup")

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

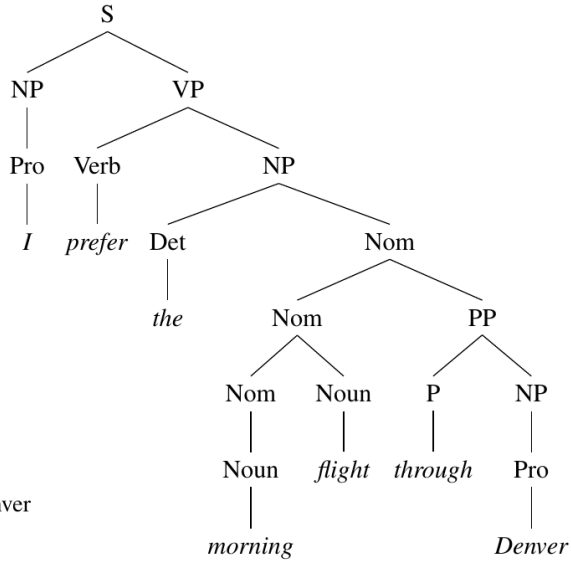


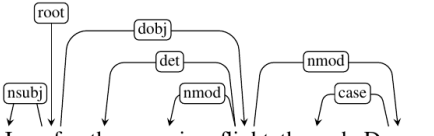
René Witte

Constituent-based Parse Tree vs. Dependency Parsing

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







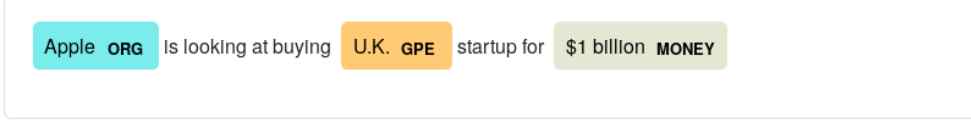
→ **Worksheet #8: Task 6**

René Witte

Named Entity Recognition 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 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.



René Witte

Entity Models

Which entities are detected?

- Depends on the model and its training data
- E.g., spaCy trained on the OntoNotes-5.0 corpus (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.)

René Witte
Concordia University

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 effort for feature engineering, training, evaluation, etc.

René Witte
Concordia University

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**

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 striking the match against a suitable surface

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 provides an API for linking entities to a knowledge base, but (currently) no pre-trained models.

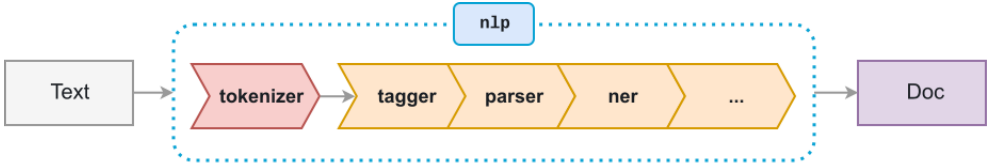
Example Pipeline

See some experimental development code using Wikidata at 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 (Easte
```

René Witte

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.

René Witte

René Witte
Concordia University

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)
```

See https://spacy.io/usage/processing-pipelines

Writing a spaCy component

René Witte

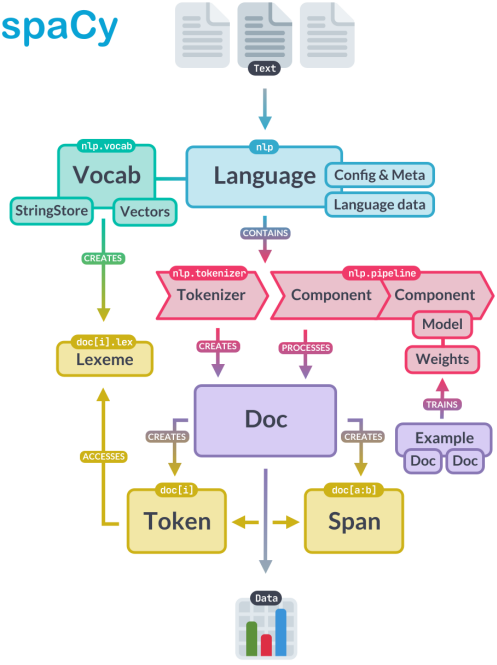
A simple "info" component

```
import spacy

def my_component(doc):
    print("After␣tokenization,␣this␣doc␣has␣{}␣tokens.".format(len(doc)))
    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(my_component, name="print_info", last=True)
print(nlp.pipe_names)
doc = nlp("This␣is␣a␣sentence.")
```

Output

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

Summary: spaCy Architecture



René Witte

Outline

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

René Witte

René Witte

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	✓

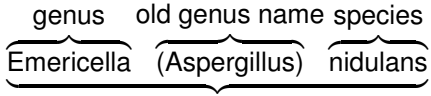
Organism

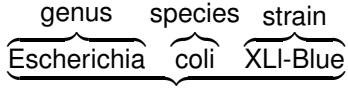
Organism	Primary Accession No. (UniProt)	Commentary	Textmining
Mus musculus	-	low-activity isozyme ADH2	-

Organism Entity Detection

René Witte

Organism Examples



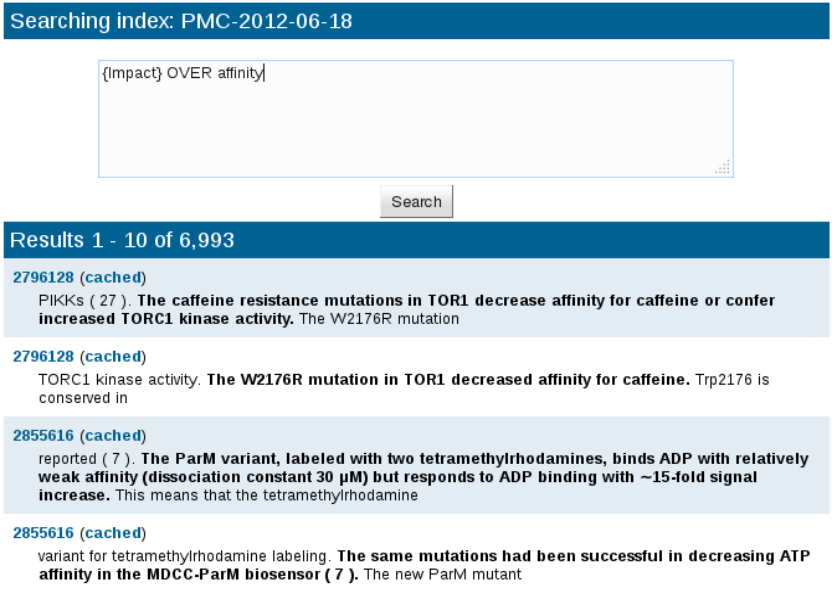


Finding Organisms: Rule matching

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)?

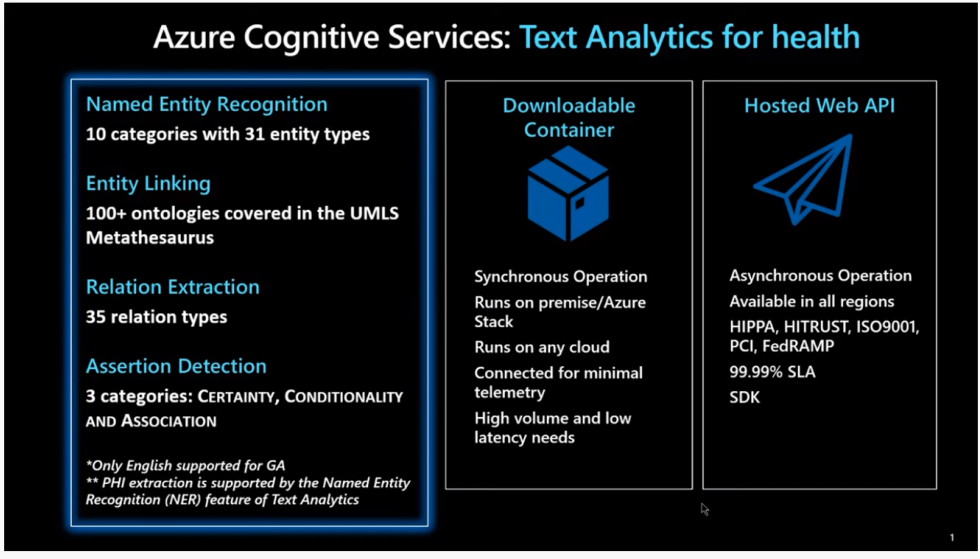
Query Interface (https://www.semanticsoftware.info/omm-query)

"Show me all protein mutations that impacted the protein property *affinity*"



[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

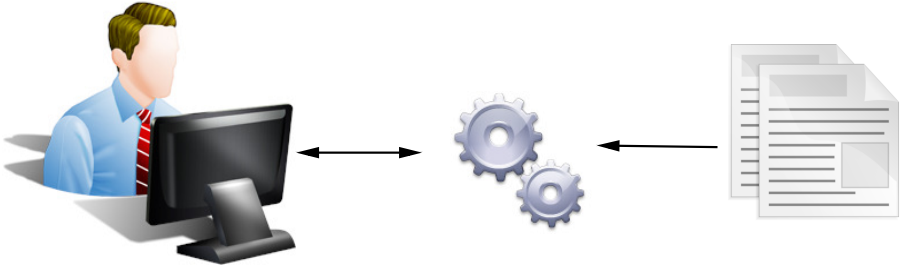
https://www.youtube.com/watch?v=_RIBf01nxI0

René Witte

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

Outline

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.