Natural Language processing (NLP) “Levels” of linguistic analysis

Thierry Poibeau (CNRS & PSL/ENS, Lattice)

thierry.poibeau@ens.psl.eu

DHAI Instensive Week, 29 March 2022

PR[AI]RIE
PaRis Artificial Intelligence Research InstitutE

ENS | PSL

cnrs

Intro (1): Natural Language Processing

- **Natural language processing** (**NLP**) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. The result is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.

https://en.wikipedia.org/wiki/Natural_language_processing

Intro (2): Natural Language Processing

- Challenges in natural language processing frequently involve speech recognition, natural language understanding, and natural-language generation

- As a summary, NLP is a super wide area of research
- Here, we will only address (a very tiny part of) natural language understanding

Levels of Linguistic Analysis

Why is NLP Hard?

Evaluation

Implementations

Conclusion

Natural Language Pyramid

Part of speech (pos) tagging

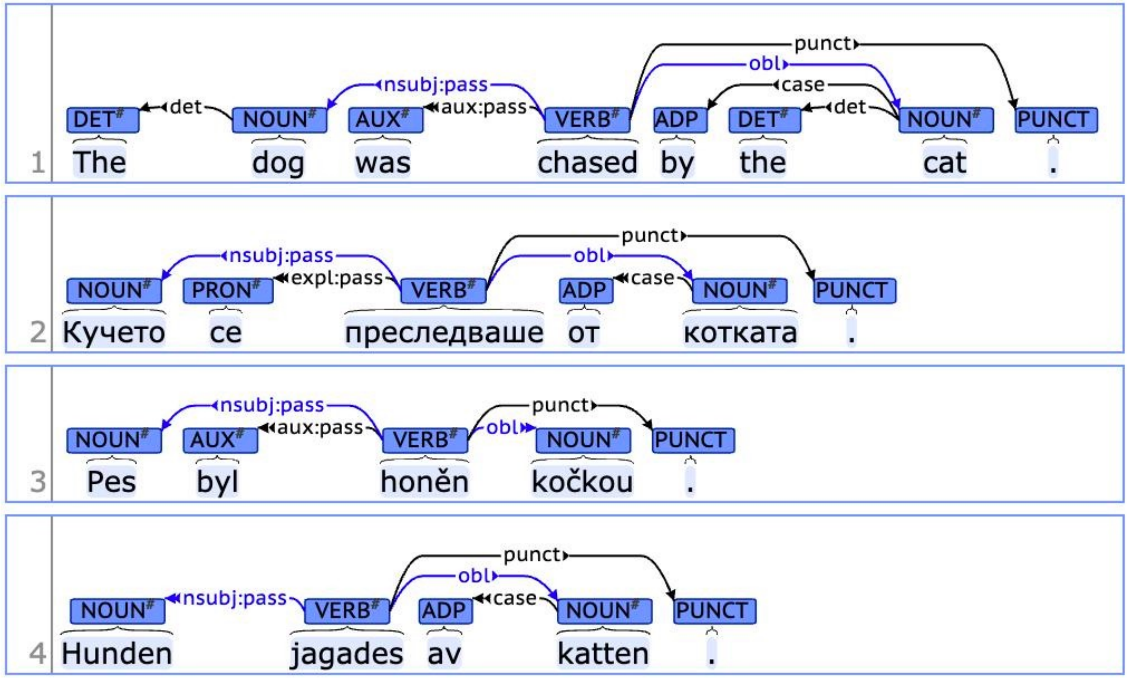
The_DT first_JJ time_NN he_PRP was_VBD shot_VBN in_IN the_DT hand_NN as_IN he_PRP chased_VBD the_DT robbers_NNS outside_RB ._.

first	time	shot	in	hand	as	chased	outside
JJ	NN	NN	IN	NN	IN	JJ	IN
RB	VB	VBD	RB	VB	RB	VBD	JJ
		VBN	RP			VBN	NN
							RB

- Lots of tools available (TreeTagger, Stanford Tagger, UD Tagger…) for numerous languages
- Accuracy (F-measure): often .9-.97 on standard text

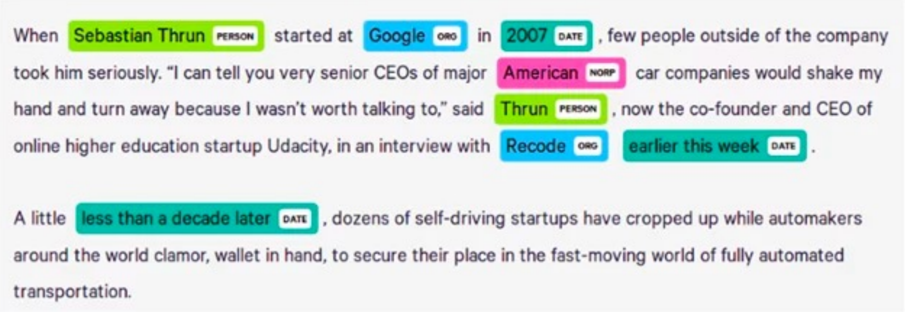
Parsing (automatic syntactic analysis)

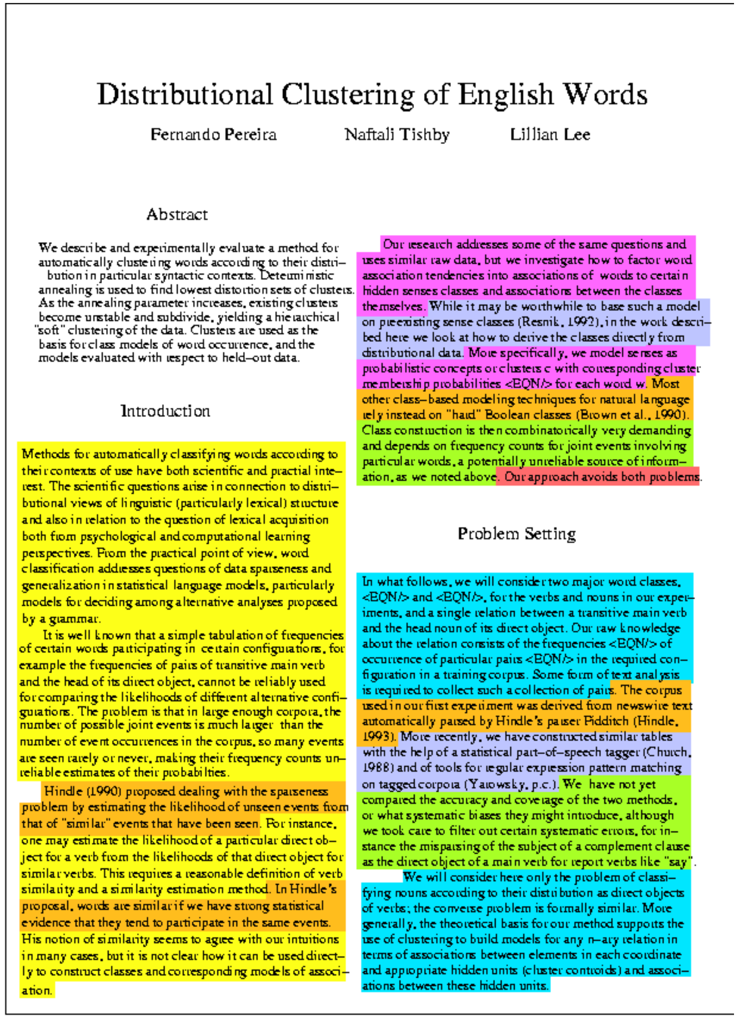
- Large diversity of tools and resources
- https://universaldependencies.org/ ~100 languages, ~200 treebanks with similar annotations
- Accuracy (F-measure): hard to predict, generally .7-.9 on standard text



Semantics

- A wide diversity of tasks
  - Word sense disambiguation (WSD)
  - Named entity recognition
  - Term recognition / Terminology
  - Text zoning
  - Event recognition
  - Sentiment / Opinion mining





Distributional Clustering of English Words

Fernando Pereira Naftali Tishby Lillian Lee

Abstract

We describe and experimentally evaluate a method for automatically clustering words according to their distribution in particular syntactic contexts. Deterministic annealing is used to find lowest distortion sets of clusters. As the annealing parameter increases, existing clusters become unstable and subdivide, yielding a hierarchical "soft" clustering of the data. Clusters are used as the basis for class models of word occurrence, and the models evaluated with respect to held–out data.

Introduction

Methods for automatically classifying words according to their contexts of use have both scientific and practial interest. The scientific questions arise in connection to distributional views of linguistic (particularly lexical) structure and also in relation to the question of lexical acquisition both from psychological and computational learning perspectives. From the practical point of view, word classification addresses questions of data sparseness and generalization in statistical language models, particularly models for deciding among alternative analyses proposed by a grammar.

It is well known that a simple tabulation of frequencies of certain words participating in certain configurations, for example the frequencies of pairs of transitive main verb and the head of its direct object, cannot be reliably used for comparing the likelihoods of different alternative configurations. The problem is that in large enough corpora, the number of possible joint events is much larger than the number of event occurrences in the corpus, so many events are seen rarely or never, making their frequency counts unreliable estimates of their probabilties.

Hindle (1990) proposed dealing with the sparseness problem by estimating the likelihood of unseen events from that of "similar" events that have been seen. For instance, one may estimate the likelihood of a particular direct object for a verb from the likelihoods of that direct object for similar verbs. This requires a reasonable definition of verb similarity and a similarity estimation method. In Hindle's proposal, words are similar if we have strong statistical evidence that they tend to participate in the same events. His notion of similarity seems to agree with our intuitions in many cases, but it is not clear how it can be used directly to construct classes and corresponding models of association.

Our research addresses some of the same questions and uses similar raw data, but we investigate how to factor word association tendencies into associations of words to certain hidden senses classes and associations between the classes themselves. While it may be worthwhile to base such a model on preexisting sense classes (Resnik, 1992), in the work described here we look at how to derive the classes directly from distributional data. More specifically, we model senses as probabilistic concepts or clusters c with corresponding cluster membership probabilities <EQN/> for each word w. Most other class–based modeling techniques for natural language rely instead on "hard" Boolean classes (Brown et al., 1990). Class constuction is then combinatorically very demanding and depends on frequency counts for joint events involving particular words, a potentially unreliable source of information, as we noted above. Our approach avoids both problems.

Problem Setting

In what follows, we will consider two major word classes, <EQN/> and <EQN/>, for the verbs and nouns in our experiments, and a single relation between a transitive main verb and the head noun of its direct object. Our raw knowledge about the relation consists of the frequencies <EQN/> of occurrence of particular pairs <EQN/> in the required configuration in a training corpus. Some form of text analysis is required to collect such a collection of pairs. The corpus used in our first experiment was derived from newswire text automatically parsed by Hindle's parser Fidditch (Hindle, 1993). More recently, we have constructed similar tables with the help of a statistical part–of–speech tagger (Church, 1988) and of tools for regular expression pattern matching on tagged corpora (Yarowsky, p.c.). We have not yet compared the accuracy and coverage of the two methods, or what systematic biases they might introduce, although we took care to filter out certain systematic errors, for instance the misparsing of the subject of a complement clause as the direct object of a main verb for report verbs like "say".

We will consider here only the problem of classifying nouns according to their distribution as direct objects of verbs; the converse problem is formally similar. More generally, the theoretical basis for our method supports the use of clustering to build models for any n–ary relation in terms of associations between elements in each coordinate and appropriate hidden units (cluster centroids) and associations between these hidden units.

From Simone Teufel (2006)

Applications

- Spell and grammatical checker
- Search engine
- Information extraction
- Text summarization
- Machine translation
- Dialogue, conversational agents
- Opinion mining
- etc.

Levels of Linguistic Analysis

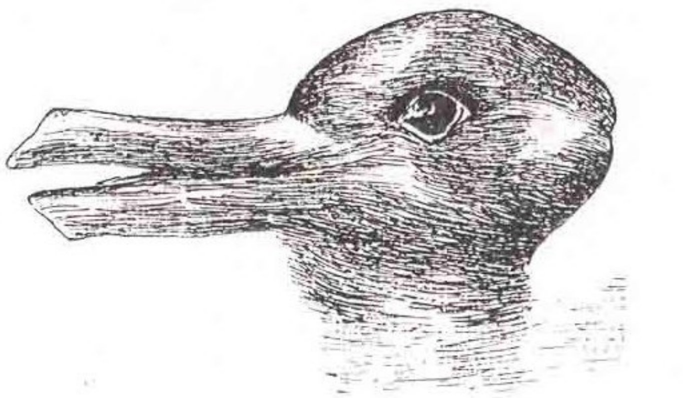
Why is NLP Hard?

Evaluation

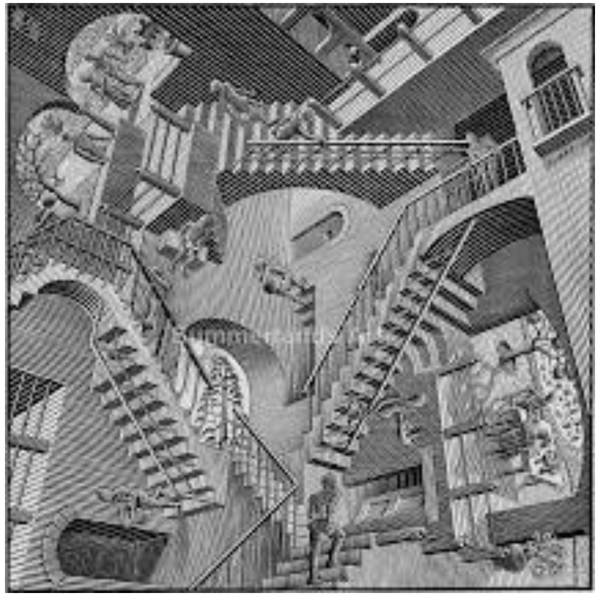
Implementations

Conclusion

Why is NLP hard?

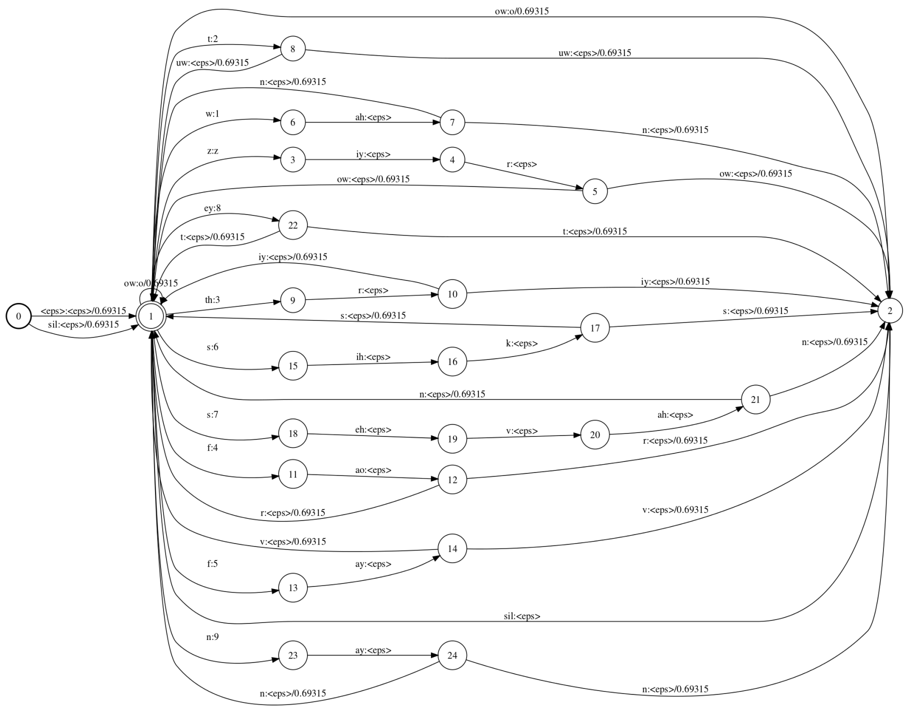


- Words are polysemous, language is ambiguous
    - Il a free, il a tout compris.
    - I buried $100 in the bank.
    - Flying planes are dangerous.
    - We saw her duck (R. Nordquist)
- Not only a linguist' / an artificial problem. Cf "*Good*"
    - "useful" or "functional" (*That's a good hammer*)
    - "exemplary" (*She's a good student*),
    - "pleasing" (*This is good soup*),
    - "moral" (*a good person*)
    - *"*righteous (*I have a good daughter*)



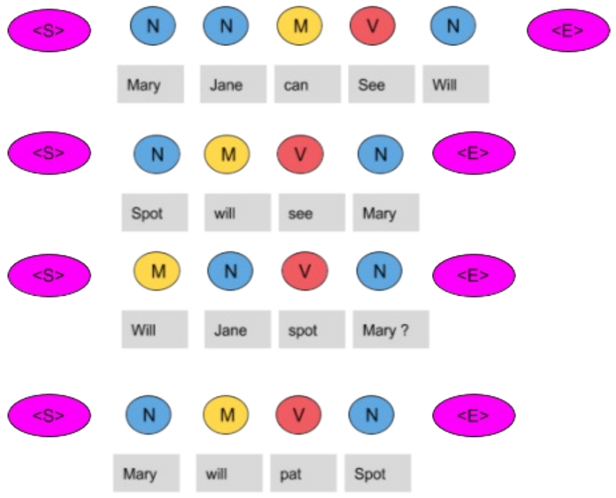
How does it work? Rule-based systems

- 1950-1990: Rule-based systems
  - Dictionary + grammar
  - Finite state transducers
- Benefits
  - Easy to read and develop
  - “Naturalness” of the approach
- Limitations
  - Poor coverage
  - Hard to maintain
  - Unsuitable for some task (WSD)



How does it work? Statistical systems

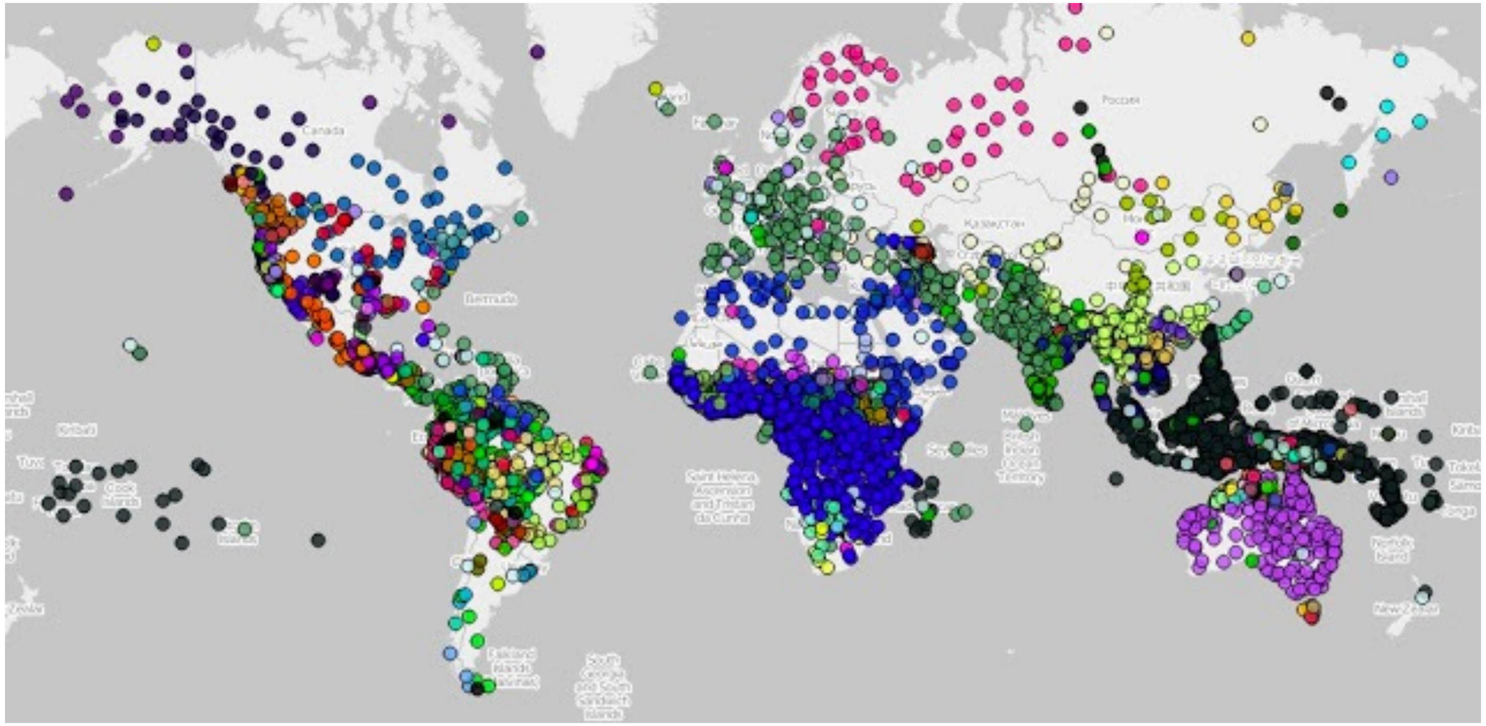
- 1990-2014: Statistical systems
  - Learn a model from representative data
  - Apply it to new data
- Benefits
  - Good coverage
  - Takes into account the statistical nature of language
- Limitations
  - Needs annotated data
  - Takes into account only the local context



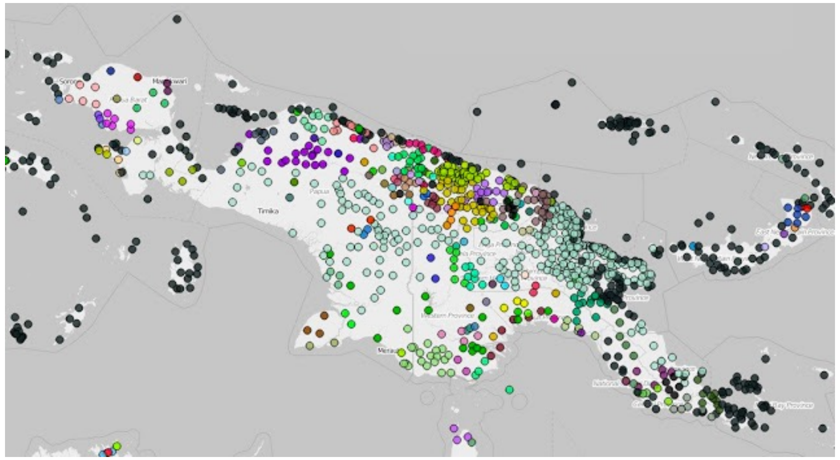
How does it work? Deep learning

- 2014-: Deep learning systems
  - A continuation of the previous approach
- Benefit
  - Better coverage, better generalizations
  - Takes into account larger contexts (transformers)
  - Universal representation (through vectors)
- Limitations
  - Needs even more annotated data
  - Exact nature of the generalizations unknown
  - Not fully reliable

Beware of Language Diversity!



http://humans-who-read-grammars.blogspot.com/2017/06/world-map-of-language-families-from.html



New Guinea

- 7000 languages in the world
- Very few of them have accurate NLP tools

Levels of Linguistic Analysis

Why is NLP Hard?

Evaluation

Implementations

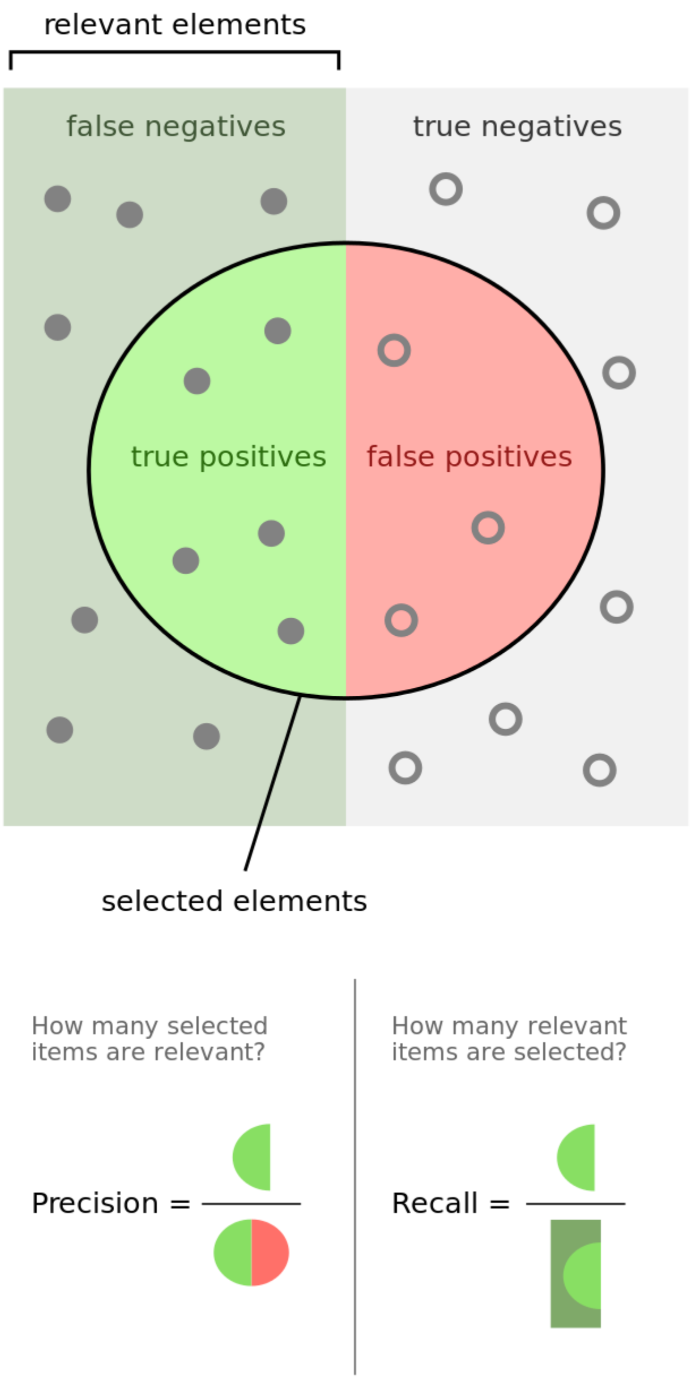
Conclusion

Overview on Evaluation

- Evaluation play a central role in NLP
  - Compare different approaches
  - Monitor progression of the field
  - Human evaluation is costly and often not so reliable
- A large number of metrics have been proposed (a research area in itself)
  - Bleu for Machine translation, Rouge for summarization, etc.
- However, most NLP tasks can be evaluated using precision and recall

Evaluation: Precision and Recall

- For any element in the dataset
  - Have all the relevant elements been tagged?
  - Are all the tagged elements relevant?
- Evaluation indicators
  - Precision: # relevant tagged element /# tagged elements
  - Recall: # relevant tagged element /# elements to be tagged
  - F-measure = 2 * (P * R) / P + R



Source : Wikipedia, https://en.wikipedia.org/wiki/Precision_and_recall

Levels of Linguistic Analysis
Why is NLP Hard?
Evaluation
Implementations
Conclusion

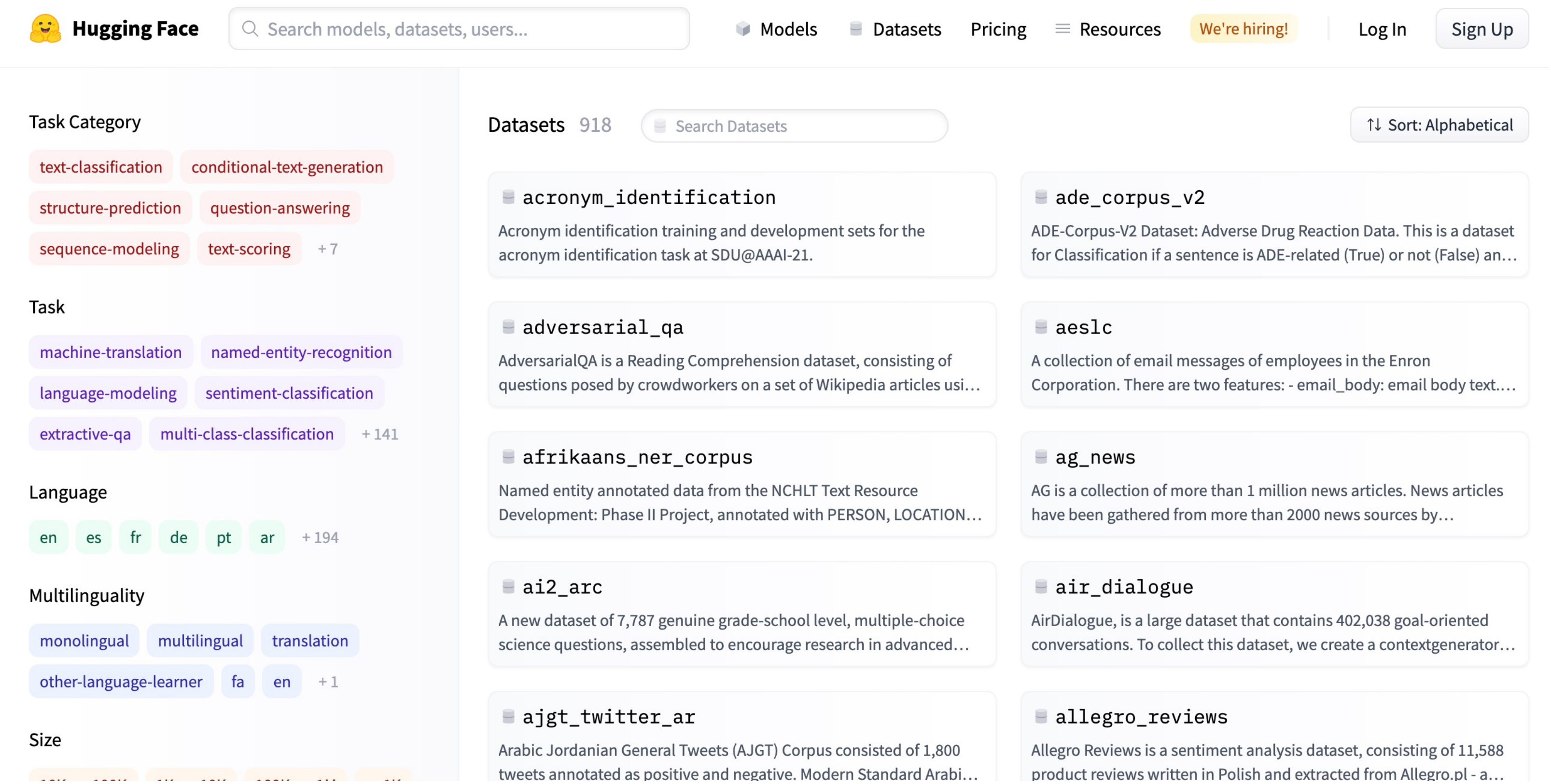
Lots of resources available!

- Lots of data, corpora and open source code available online (remember UD)
- Most recent models are open source and easy to integrate (including models from companies like Facebook, cf. FastText, or Google, cf. Bert)
- Deep learning means GPU are requited (Google Colab can help)

- But also note: there are lots of languages with few or nearly no resources!   (~7000 languages in the world)

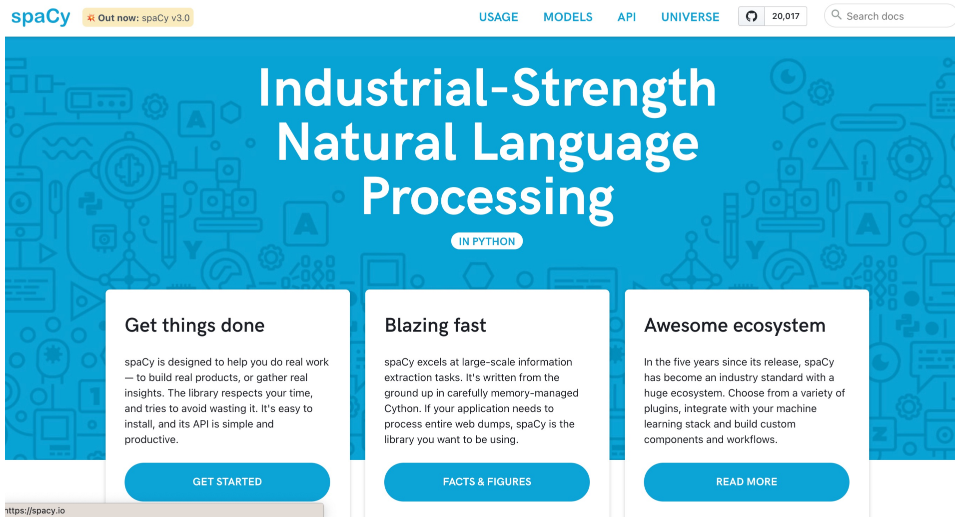
Hugging Face

- Recent developments (Transformers, cf. Bert in different languages etc. https://huggingface.co/



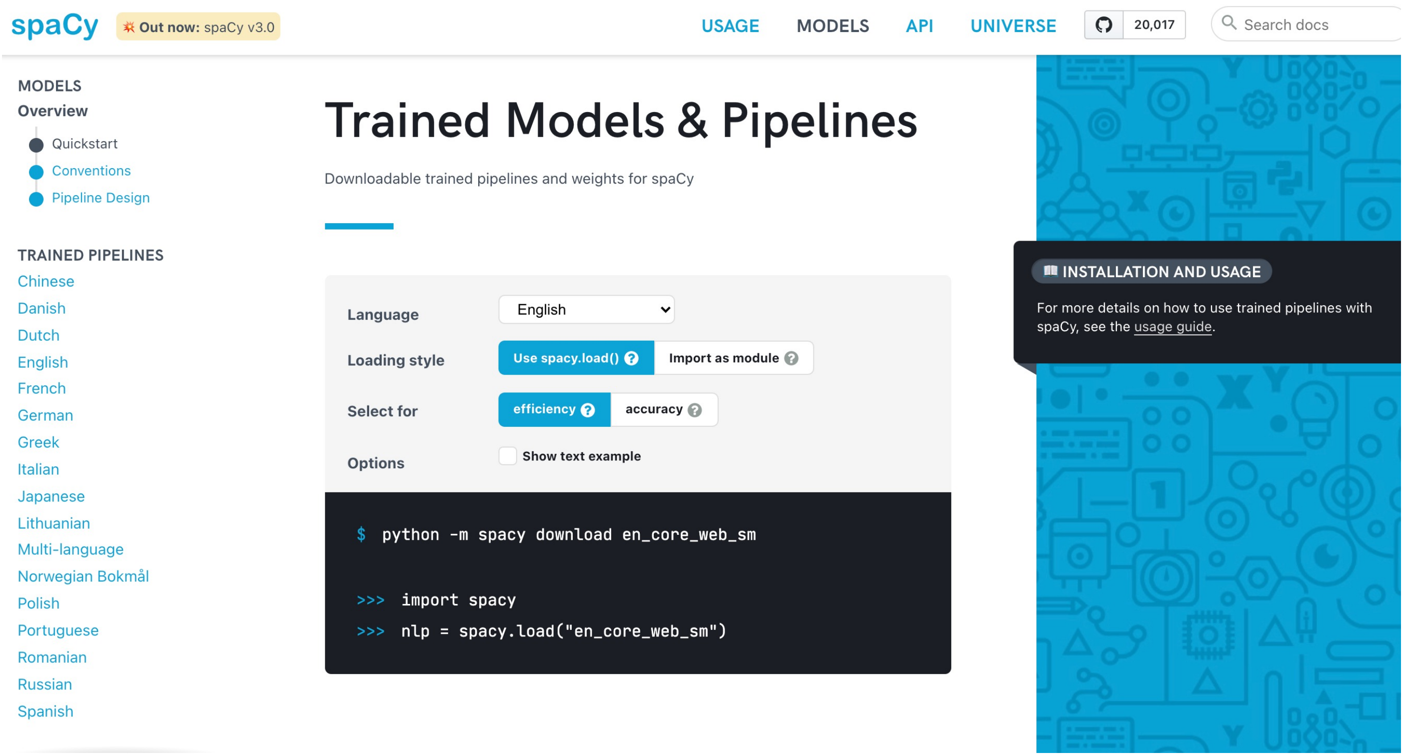
Spacy

- General NLP: Spacy (https://spacy.io/)
- Integrates Huggingface work on recent large scale NLP models
- Easier to use than directly manipulating Hugging Face code



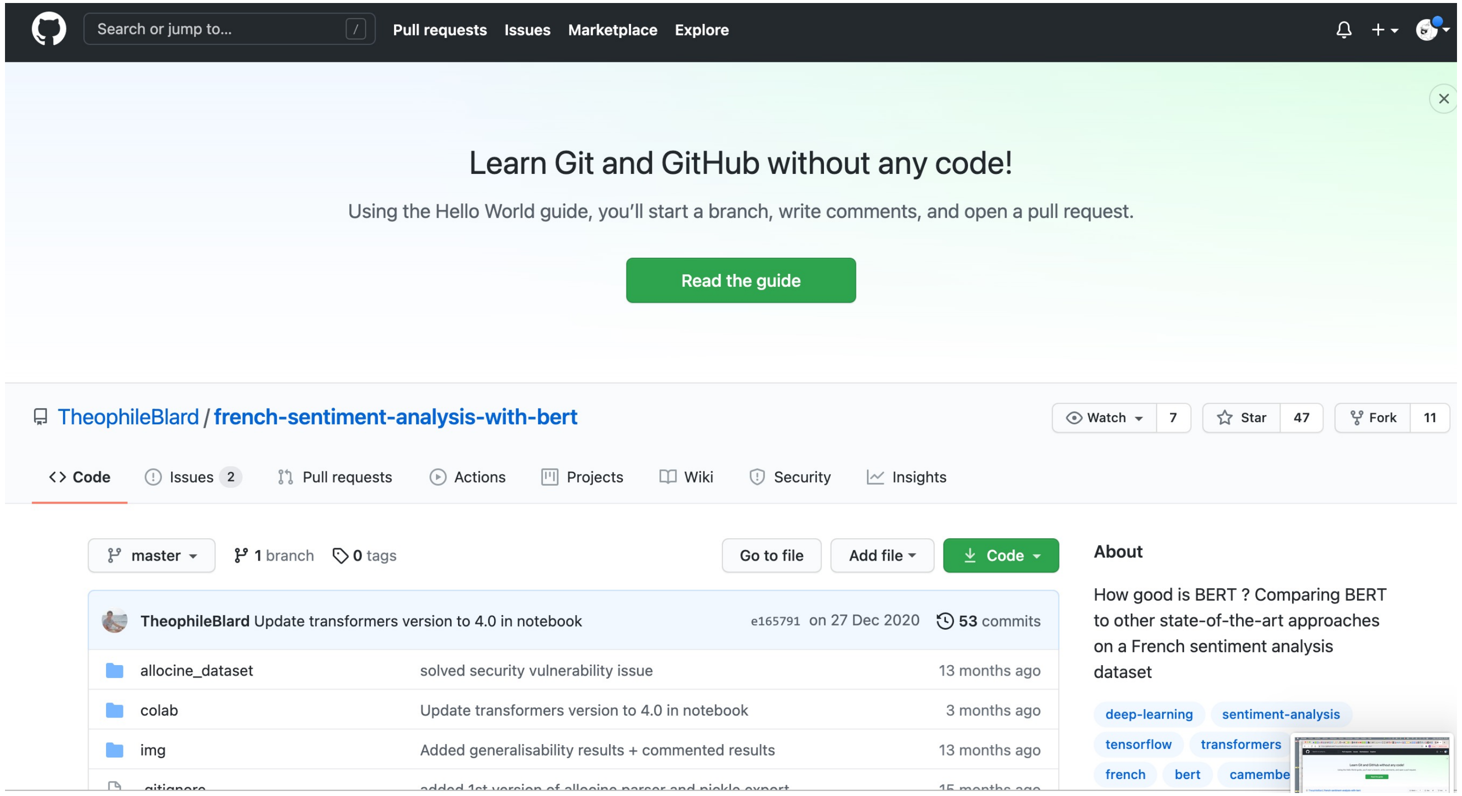
Spacy

- Models for different languages



Github / Jupyter Notebooks

- Jupyter notebooks, for example Sentiment analysis in French on a corpus made of Allocine reviews



Named entity recognition using Spacy

jupyter tutorielNERSpacy Dernière Sauvegarde : 20/02/2021 (auto-sauvegardé)

Se déconnecter

Fichier Édition Affichage Insérer Cellule Noyau Widgets Aide

Fiable | Python 3

Exécuter Markdown

Entrée [ ]:
```
import spacy
nlp = spacy.load("fr_core_news_md")
```

2. Le code suivant permet de lire un fichier texte founi en entrée, ici il s'agit du premier chapitre de Nana (Zola).

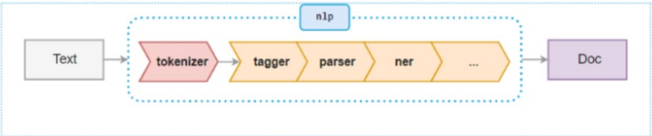
Entrée [ ]:
```
with open ("Nana/chapitre1.txt", "r", encoding="utf_8") as myfile:
    data = myfile.read()
```

3. Nous allons appliquer le modèle REN sur le texte contenu dans la variable data (cela peut prendre quelques minutes).

Entrée [ ]:
```
doc = nlp(data)
```

4. Le code suivant permet d'afficher ce que l'on retrouve dans la variable Doc.

Entrée [ ]:
```
print(type(doc))
```



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

5. Nous allons imprimer la liste d'EN trouvées avec la catégorie et l'information des index de caractères dans le texte (offset).

Entrée [5]:
```
def show_ents(doc):
    if doc.ents:
        for ent in doc.ents:
            print(ent.text+' - ' +str(ent.start_char) +' - '+ str(ent.end_char) +' - '+ent.label_+ ' - '+str(spacy.exp
            print("\n")
show_ents(doc)
```

```
I - 0 - 1 - LOC - Non-GPE locations, mountain ranges, bodies of water

Variétés - 42 - 50 - LOC - Non-GPE locations, mountain ranges, bodies of water

Hector - 999 - 1005 - PER - Named person or family.

Oh ! - 1171 - 1175 - MISC - Miscellaneous entities, e.g. events, nationalities, products or works of art
```

Jupyter notebook, adapted from C. Brando, Master HN PSL

Levels of Linguistic Analysis

Why is NLP Hard?

Evaluation

Implementations

Conclusion

Conclusion

- A field that evolve quickly
- Lots of progress, but lots of challenges remaining
- Bigger may not always be better

- Keep in mind language diversity, and under resource languages
- Keep in mind ethical issues (e.g. gender bias in language models)

To go further…

- Check the Bible: https://web.stanford.edu/~jurafsky/slp3/ (free!)

Speech and Language Processing (3rd ed. draft)

Dan Jurafsky and James H. Martin

Here's our December 30, 2020 draft! Includes:

- new version of Chapter 8 (bringing together POS and NER in one chapter),
- new version of Chapter 9 (with Transformers)
- Chapter 11 (MT)
- neural span parsing and CCG parsing moved into Chapter 13 (Constituency Parsing) and Statistical Constituency Parsing moved to Appendix C
- new version of Chapter 23 (QA modernized)
- Chapter 26 (ASR + TTS)
- Plus a modernizing pass (and typo fixing, thanks to all of you!!!) on all the other chapters.

We are really grateful to all of you for finding bugs and offering great suggestions!

Individual chapters are below; here is a single pdf of all the chapters in the December 30, 2020 draft of the book-so-far

As always, typos and comments very welcome (just email slp3edbugs@gmail.com and let us know the date on the draft)!
(Due to reorganizing, still expect some missing latex cross-references throughout the pdfs, don't bother reporting those missing ref/typos.)

Feel free to use the draft slides in your classes.
We are in the process of updating the slides now; so far the slides for Chapters 2, 3, 4, 5, 6, 20, and 24 have been updated.

When will the whole book be finished?
We're shooting for well before the end of 2021 for the 3 remaining chapters (Intro, Contextual Embeddings, Semantic Parsing) + random missing sections.

And if you need last year's draft chapters, they are here.

	Chapter	Slides	Relation to 2nd ed.
1:	Introduction		[Ch. 1 in 2nd ed.]
2:	Regular Expressions, Text Normalization, Edit Distance	2: Text Processing [pptx] [pdf] 2: Edit Distance [pptx] [pdf]	[Ch. 2 and parts of Ch. 3 in 2nd ed.]
3:	N-gram Language Models	3: N-grams [pptx] [pdf]	[Ch. 4 in 2nd ed.]
4:	Naive Bayes and Sentiment Classification	4: Naive Bayes + Sentiment [pptx] [pdf]	[new in this edition]