POLITÉCNICA
"Ingeniamos el futuro"

CAMPUS
DE EXCELENCIA
INTERNACIONAL

Text Mining 5
Language Processing

Madrid Summer School on
Advanced Statistics and Data Mining

Florian Leitner
Data Catalytics, S.L.
leitner@datacatytics.com

Evaluation metrics for classification tasks

Evaluations should answer questions like:

How to measure a change to an approach?

Did adding a feature improve or decrease performance?

Is the approach good at locating the relevant pieces or good at excluding the irrelevant bits?

How do two or more different methods compare?

Essential evaluation metrics: Accuracy, F-Measure, MCC Score

Patient→ Doctor↓	has cancer	is healthy
diagnose cancer	TP	FP
detects nothing	FN	TN

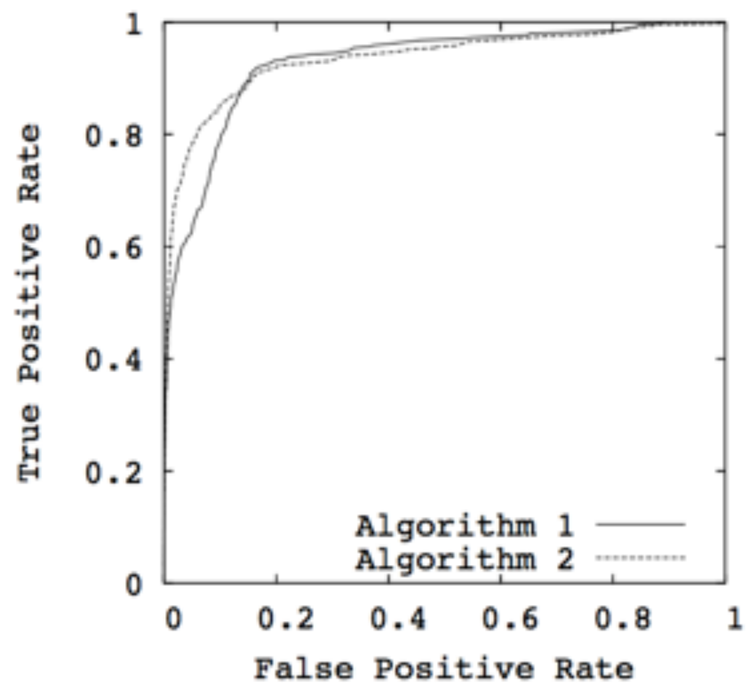
- **Precision** (P)
  - correct hits [TP] ÷ all hits [TP + FP]
- **Recall** (R; **Sensitivity**, TPR)
  - correct hits [TP] ÷ true cases [TP + FN]
- **Specificity** (True Negative Rate)
  - correct misses [TN] ÷ negative cases [FP + TN]

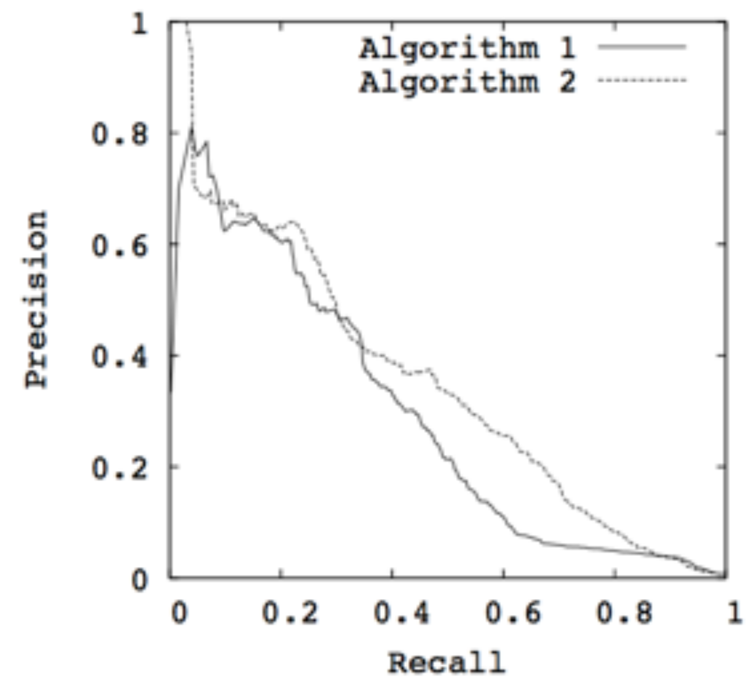
NB: no result order!

- **Accuracy**
  - correct classifications [TP + TN] ÷ all cases [TP + TN + FN + FP])
  - highly **sensitive to** class **imbalance**
- **F-Measure** (F-Score)
  - the harmonic mean between P & R = 2 TP ÷ (2 TP + FP + FN) = (2 P R) ÷ (P + R)
  - does **not** require a **TN** count
- **MCC Score** (Mathew's Correlation Coefficient)
  - **χ^2-based**: (TP TN - FP FN) ÷ sqrt[(TP+FP)(TP+FN)(TN+FP)(TN+FN)]
  - **robust against** class **imbalance**

Ranked evaluation results: AUC ROC and PR

Area Under the Curve
Receiver-Operator Characteristic
Precision-Recall



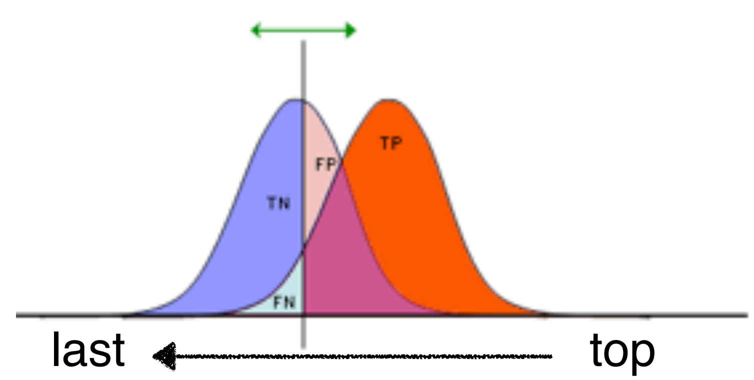


Davis & Goadrich.
ICML 2006

TPR / Recall *(aka. Sensitivity)*
TP ÷ (TP + FN)

FPR *(not Specificity!)*
FP ÷ (TN + FP)

Precision
TP ÷ (TP + FP)



TP	FP
FN	TN
1	1

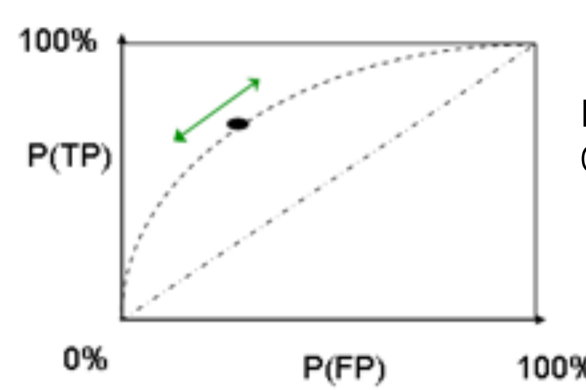


Image Source: WikiMedia Commons, kku (“kakau”, eddie)

To ROC or to PR?

Curve I:
10 hits in the top 10, and 10 hits spread over the next 1500 results.

AUC ROC 0.813

Results: 20 T ≪ 1980 N

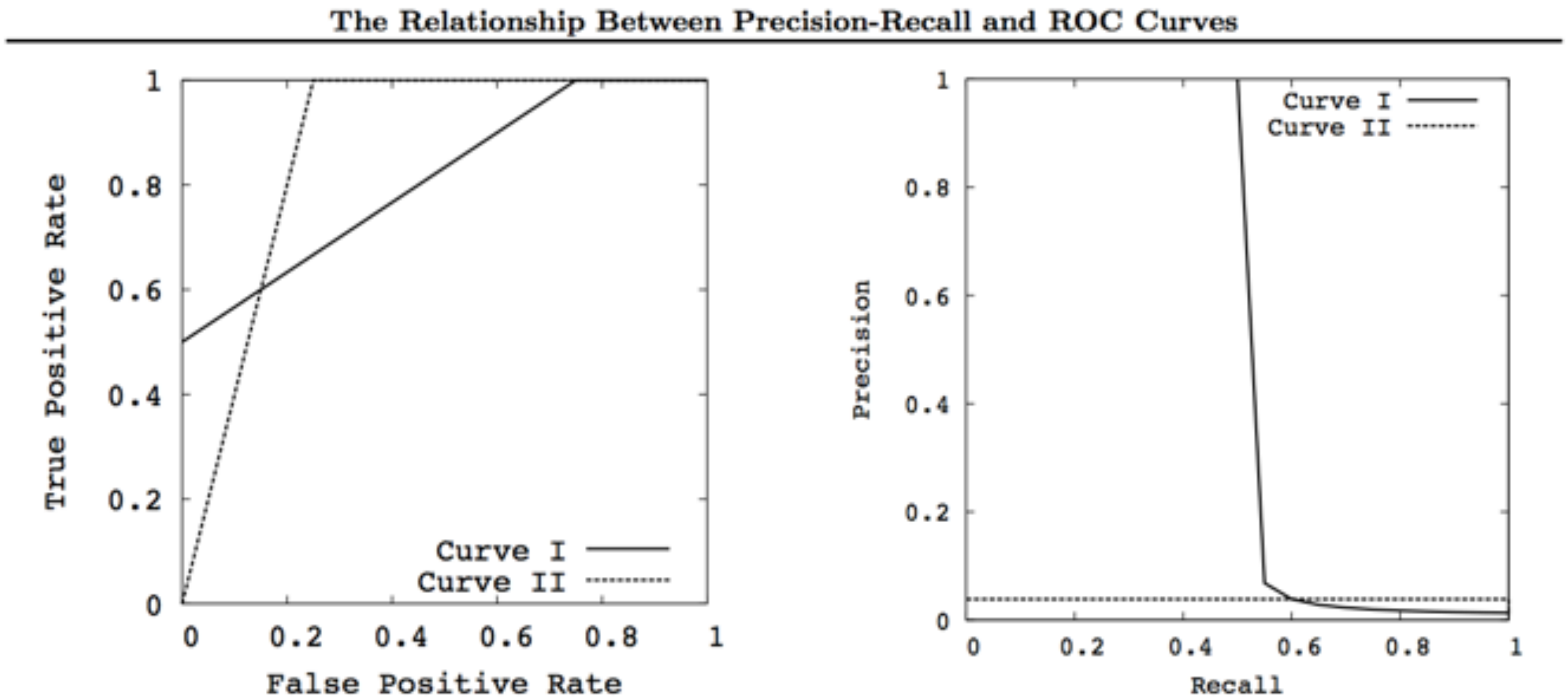


Figure 11. Comparing AUC-ROC for Two Algorithms

Figure 12. Comparing AUC-PR for Two Algorithms

Curve II:
Hits spread evenly over the first 500 results.

AUC ROC 0.875

"An algorithm which optimizes the area under the ROC curve is not guaranteed to optimize the area under the PR curve."

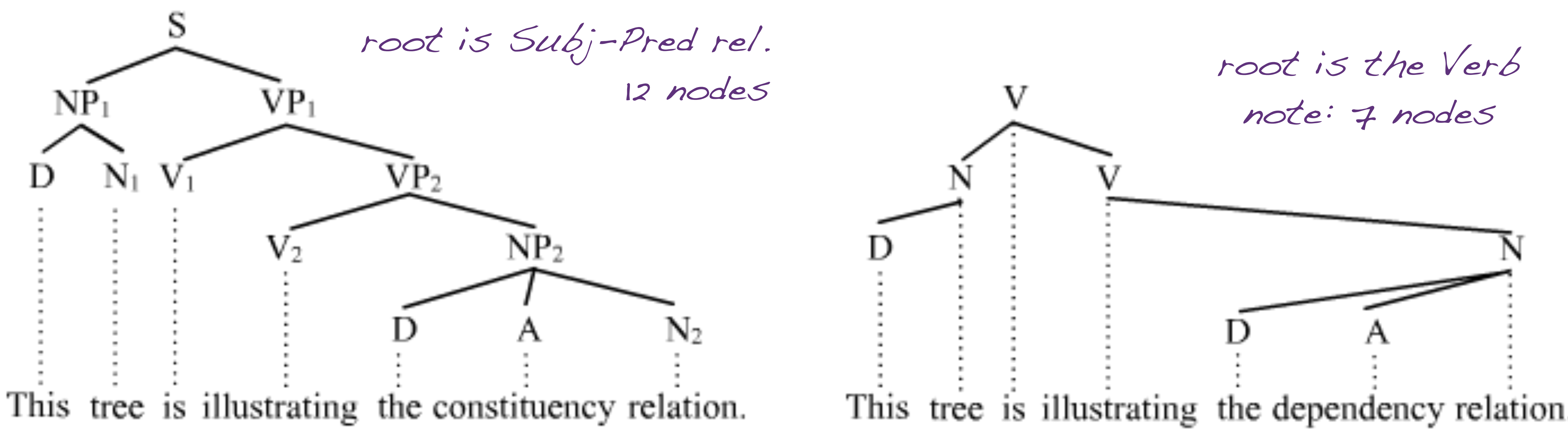
Davis & Goadrich, 2006

- Davis & Goadrich. The Relationship Between PR and ROC Curves. ICML 2006
- Landgrebe et al. Precision-recall operating characteristic (P-ROC) curves in imprecise environments. Pattern Recognition 2006
- Hanczar et al. Small-Sample Precision of ROC-Related Estimates. Bioinformatics 2010

→ Use (AUC) PR for [imbalanced] ranking scenarios!

Detecting grammatical (sentence) structure

Phrase-structure (aka. **constituency**) vs. **dependency** grammars



Constituency relation (PSG)

Dependency relation

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

P-S Grammars: **Chomsky**; Dependency Grammars: **Tesnière**

Dependency relations can be annotated with a linear-time parser.

note the one-to-many constituency vs. the one-to-one dependency relations

Tesnière's dependency relations (1959)

He ate pizza with anchovies. He ate pizza with friends.

ate(he, pizza with anchovies)
~~ate(he, with anchovies)~~

Relationships

ate(he, pizza)
ate(he, with friends)

NB: Dependencies cannot capture **phrasal structure** (subject, object, verb phrase, etc.), and in particular, **word order**.

which can be a benefit: some languages have a free word order, e.g. Turkish or Czech
reminder: clauses and collocations are special phrasal structures

Dependency parsing 1/2

- Transition-based, arc-standard, shift-reduce, greedy parsing.
- The default approach to dependency parsing today is O(n).
  - **Transition-based**: Move from one token to the next.
  - **Arc-standard**: assign arcs when the dependent token (at the arrowhead) is fully resolved (common alternative: arc-eager → assign the arcs immediately).
  - **Shift-reduce**: A stack of words and a stream buffer: either shift next word from the buffer to the stack or reduce a word from the stack by “arcing”.
  - **Greedy**: Make locally optimal transitions (assume independence of arcs).

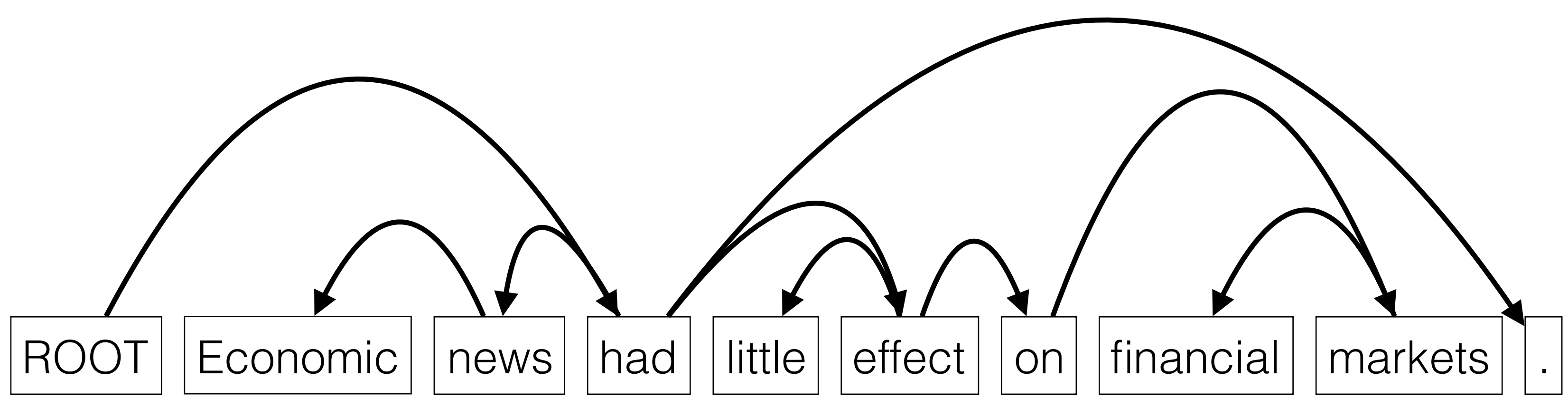
A shift-reduce parse

Buffer Economic news had little effect on financial markets .

Stack
.
markets
financial
on
effect
little
had
news
Economic
ROOT

Shift: Economic
Shift: news
Left-arc: Economic←news
Shift: had
Left-arc: news←had
Shift: little
Shift: effect
Left-arc: little←effect
Shift: on

Shift: financial
Shift: markets
Left-arc: financial←markets
Right-arc: on→markets
Right-arc: effect→on
Right-arc: had→effect
Shift: .
Right-arc: had→.
Right-arc: ROOT→had



Dependency Parsing. Kübler et al., 2009

A shift-reduce parse

Shift: Economic

Shift: news

Left-arc: Economic←news

Shift: had

Left-arc: news←had

Shift: little

Shift: effect

Left-arc: little←effect

Shift: on

Shift: financial

Shift: markets

Left-arc: financial←markets

Right-arc: on→markets

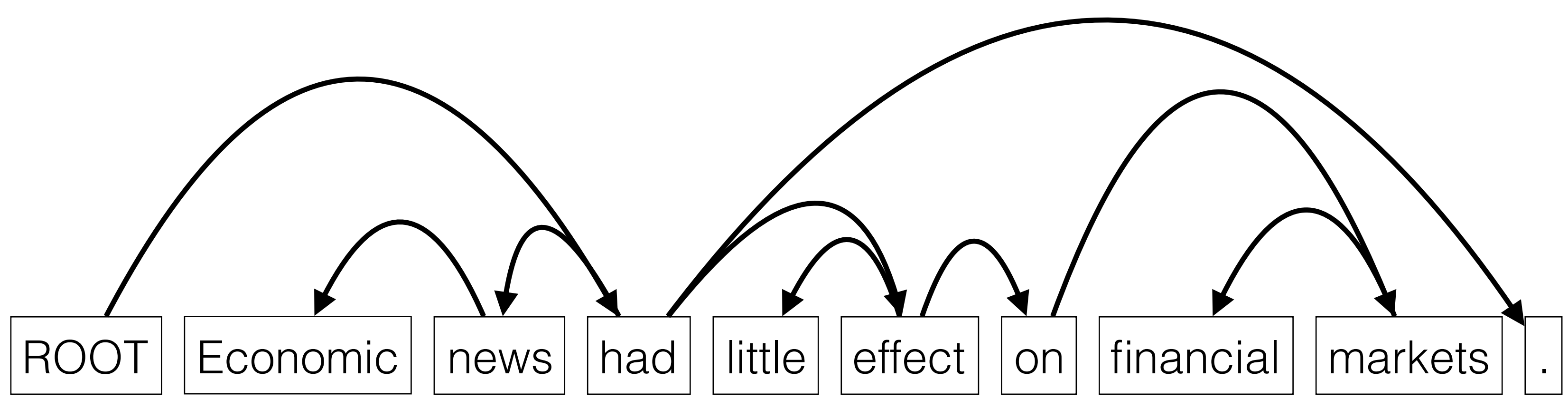
Right-arc: effect→on

Right-arc: had→effect

Shift: .

Right-arc: had→.

Right-arc: ROOT→had



Dependency Parsing. Kübler et al., 2009

Dependency parsing 2/2

- (Arc-standard) Transitions: **shift** or **reduce** (left-arc, right-arc)
- Transitions are chosen using some classifier
  - Maximum entropy classifier, support vector machine, single-layer perceptron, perceptron with one hidden layer (→ Stanford parser, 2014 edition, SpaCy v1), more complex deep nets (→ Google's SyntaxNet, SpaCy v2)
- Main issues:
  - Few large, well annotated training corpora ("dependency **treebanks**"). Biomedical domain: GENIA; Newswire: WSJ, Prague, Penn, …
  - **Non-projective** trees (i.e., trees with arcs crossing each other; common in a number of other languages, e.g. German) with arcs that have to be drawn between nodes that are not adjacent on the stack.

Four approaches to relationship extraction

- **Co-mention window**
  - E.g.: if ORG and LOC entity within same sentence and no more than x tokens in between, treat the pair as a hit.
  - Low precision, high recall; trivial, many false positives.
- **Dependency parsing**
  - If a path covering certain nodes (e.g. prepositions like "in/IN" or predicates [~verbs]) connects two entities, extract that pair.
  - Balanced precision and recall, computationally expensive.
- **Pattern extraction** (over the seg. tags)
  - e.g.: <ORG>+ <IN> <LOC>+ (preposition)
  - High precision, low recall; cumbersome, but very common.
  - Pattern **learning** can help.
- **Machine Learning**
  - Features for sentences with entities and some classifier (e.g., SVM, neural net, MaxEnt, Bayesian net, …)

    token-distance, num. of tokens between the entities, tokens before/after them, etc.
  - Highly variable milages.

    … but loads of fun in your speaker's opinion :)