NLP Project Full Circle

Vsevolod Dyomkin

prj-nlp

2020-03-26

Topics

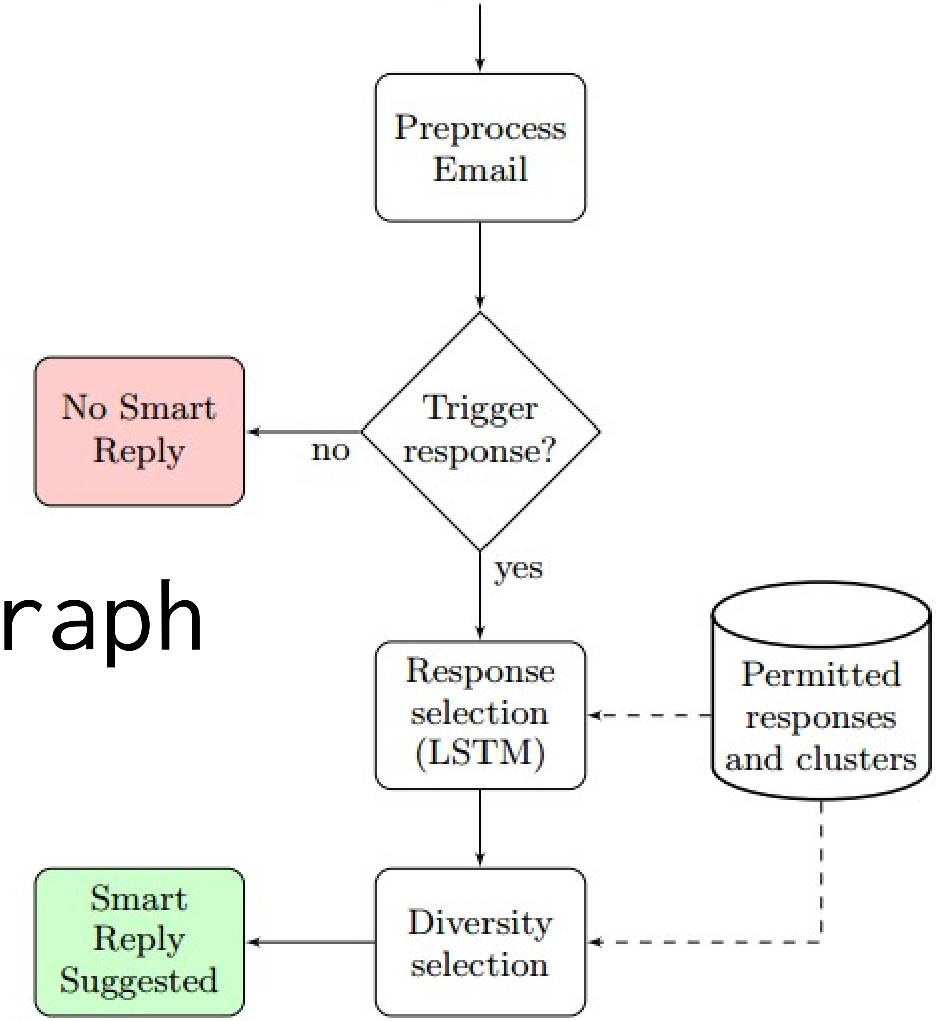
* How the project is structured
* Experiment setup
* Evaluation & metrics
* Rule-based approach

Example

Gmail Smart Reply Feature

http://www.kdd.org/kdd2016/papers/files/Paper_1069.pdf

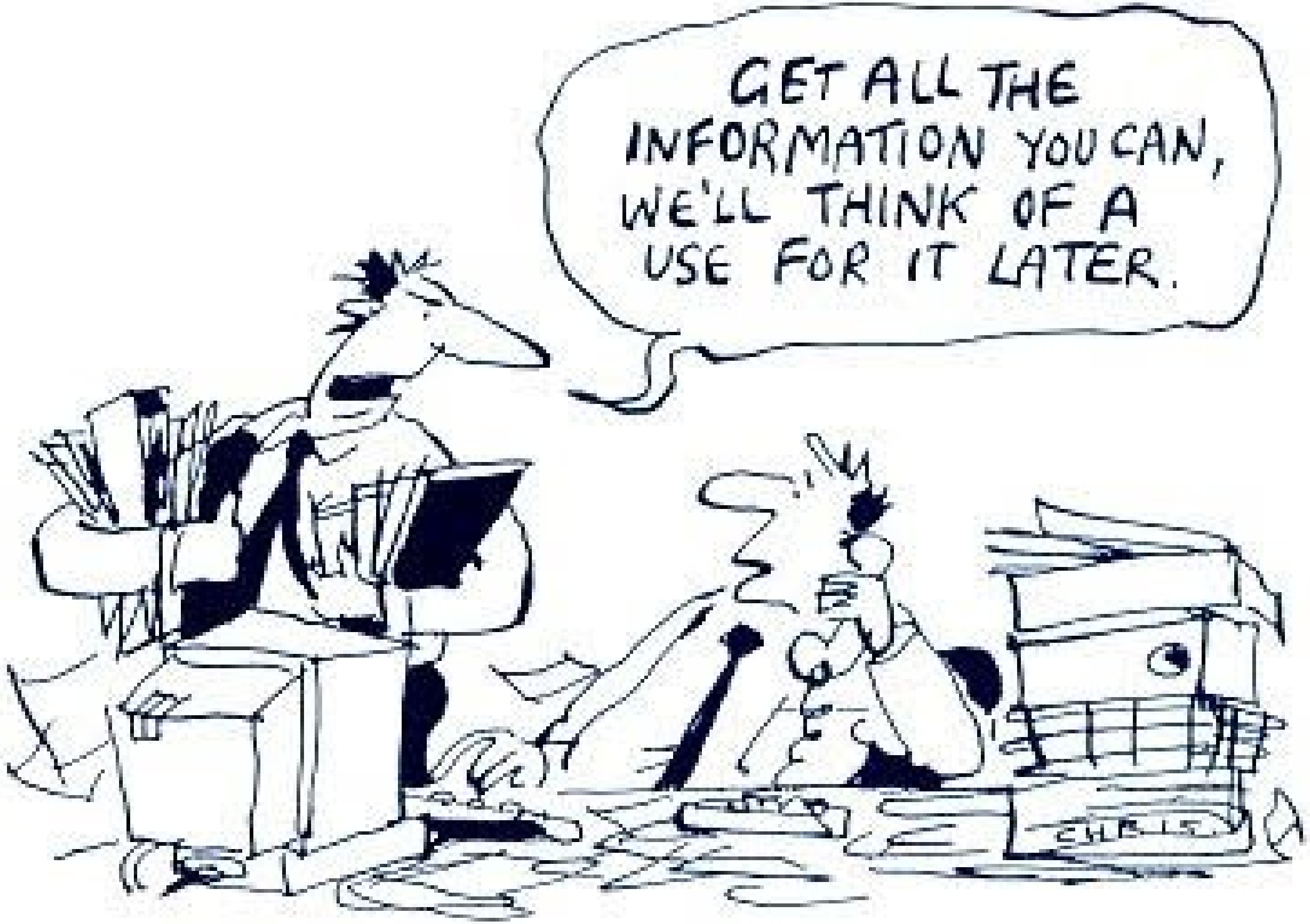
* Data analysis
* Traditional NLP processing
* FNN
* LSTM
* Semi-supervised graph learning
* Rule-based post-processing
* Engineering



NLP Project Stages

1) Domain analysis
2) Data preparation
3) Iterating the solution
4) Productizing the result
5) Gathering feedback
   and reiterating

Domain Analysis


GET ALL THE INFORMATION YOU CAN, WE'LL THINK OF A USE FOR IT LATER.

Domain Analysis

1) Problem formulation
2) Existing solutions
3) Possible datasets
4) Identifying challenges
5) Metrics, baseline & SOTA
6) Selecting possible approaches and devising a plan of attack

Глокая куздра штеко будланула бокра

И кудрячит бокрёнка

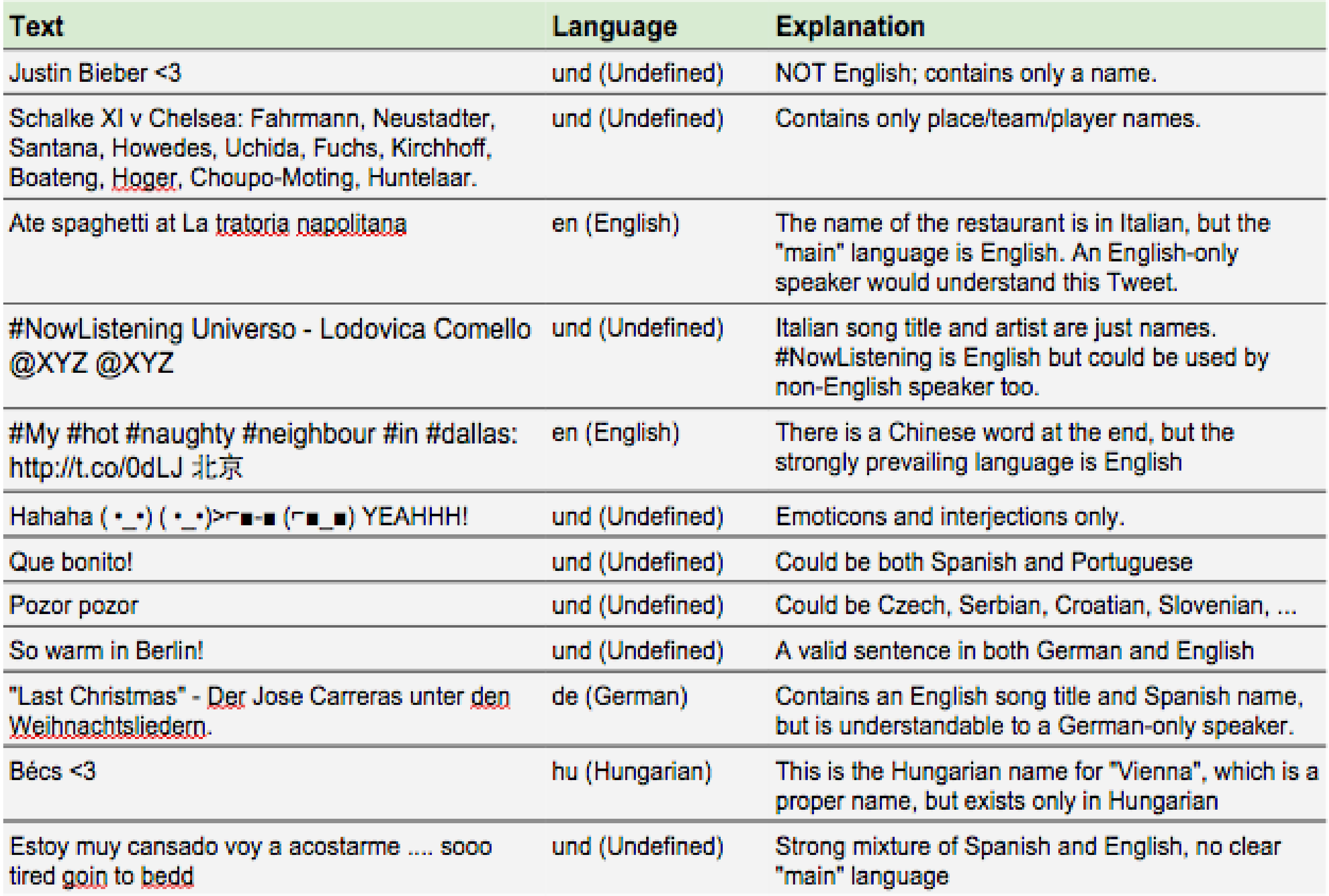
Problem Formulation

Task: language identification

* Not an unsolved problem
* Emphasis on short texts
* Cover “all” languages

* The issue of UND/MULT texts
* Dialects?
* Encodings?

Interesting Examples



Text	Language	Explanation
Justin Bieber <3	und (Undefined)	NOT English; contains only a name.
Schalke XI v Chelsea: Fahrmann, Neustadter, Santana, Howedes, Uchida, Fuchs, Kirchhoff, Boateng, Hoger, Choupo-Moting, Huntelaar.	und (Undefined)	Contains only place/team/player names.
Ate spaghetti at La tratoria napolitana	en (English)	The name of the restaurant is in Italian, but the "main" language is English. An English-only speaker would understand this Tweet.
#NowListening Universo - Lodovica Comello @XYZ @XYZ	und (Undefined)	Italian song title and artist are just names. #NowListening is English but could be used by non-English speaker too.
#My #hot #naughty #neighbour #in #dallas: http://t.co/0dLJ 北京	en (English)	There is a Chinese word at the end, but the strongly prevailing language is English
Hahaha (•_•) (•_•)>⌐■-■ (⌐■_■) YEAHHH!	und (Undefined)	Emoticons and interjections only.
Que bonito!	und (Undefined)	Could be both Spanish and Portuguese
Pozor pozor	und (Undefined)	Could be Czech, Serbian, Croatian, Slovenian, ...
So warm in Berlin!	und (Undefined)	A valid sentence in both German and English
"Last Christmas" - Der Jose Carreras unter den Weihnachtsliedern.	de (German)	Contains an English song title and Spanish name, but is understandable to a German-only speaker.
Bécs <3	hu (Hungarian)	This is the Hungarian name for "Vienna", which is a proper name, but exists only in Hungarian
Estoy muy cansado voy a acostarme .... sooo tired goin to bedd	und (Undefined)	Strong mixture of Spanish and English, no clear "main" language

https://blog.twitter.com/2015/evaluating-language-identification-performance

Existing Solutions

* https://github.com/shuyo/language-detection/ (Java)
* https://github.com/saffsd/langid.py (Python)
* https://github.com/mzsanford/cld (C++)

Existing Solutions

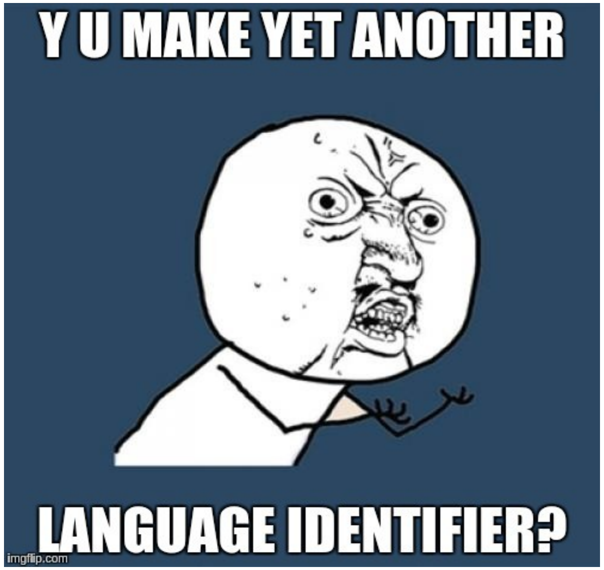
* https://github.com/shuyo/language-detection/ (Java)
* https://github.com/saffsd/langid.py (Python)
* https://github.com/mzsanford/cld (C++)

* https://github.com/CLD2Owners/cld2
* https://github.com/google/cld3

Existing Solutions

* https://github.com/shuyo/language-detection/ (Java)
* https://github.com/saffsd/langid.py (Python)
* https://github.com/mzsanford/cld (C++)

* https://github.com/CLD2Owners/cld2
* https://github.com/google/cld3



Possible Datasets

* Debian i18n (~90 langs)

Possible Datasets

* Debian i18n (~90 langs)
* TED Multilingual (109 langs)

Possible Datasets

* Debian i18n (~90 langs)
* TED Multilingual (109 langs)
* Wiktionary (~150 langs)

Possible Datasets

* Debian i18n (~90 langs)
* TED Multilingual (109 langs)
* Wiktionary (~150 langs)
* Wikipedia (175 langs)

Test Data

* Twitter evaluation dataset
* Datasets with fewer langs
* Extract from Wikipedia

* Smoke test dataset

Challenges

* Linguistic challenges
  - know next to nothing about 90% of the languages
  - languages and scripts
    http://www.omniglot.com/writing/langalph.htm
  - language distributions
    https://en.wikipedia.org/wiki/Languages_used_on_the_Internet#Content_languages_for_websites
  - word segmentation?

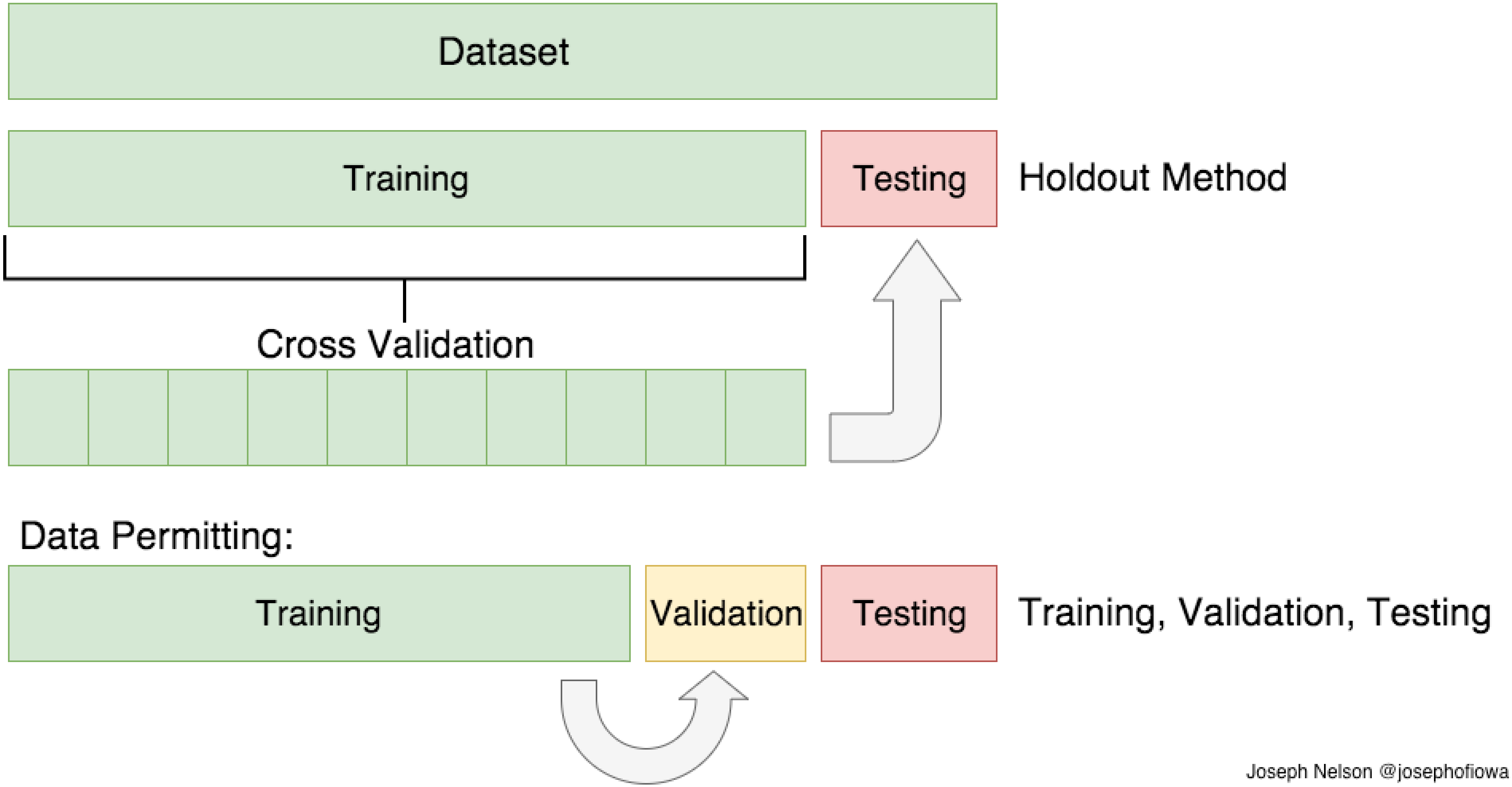
* Engineering challenges

NLP Evaluation

Intrinsic: use a gold-standard dataset and some metric to measure the system’s performance directly. (in-domain & out-of-domain)

Extrinsic: use a system as part of an upstream task(s) and measure performance change there.

Dev-Test Split



Dataset

Training

Testing

Holdout Method

Cross Validation

Data Permitting:

Training

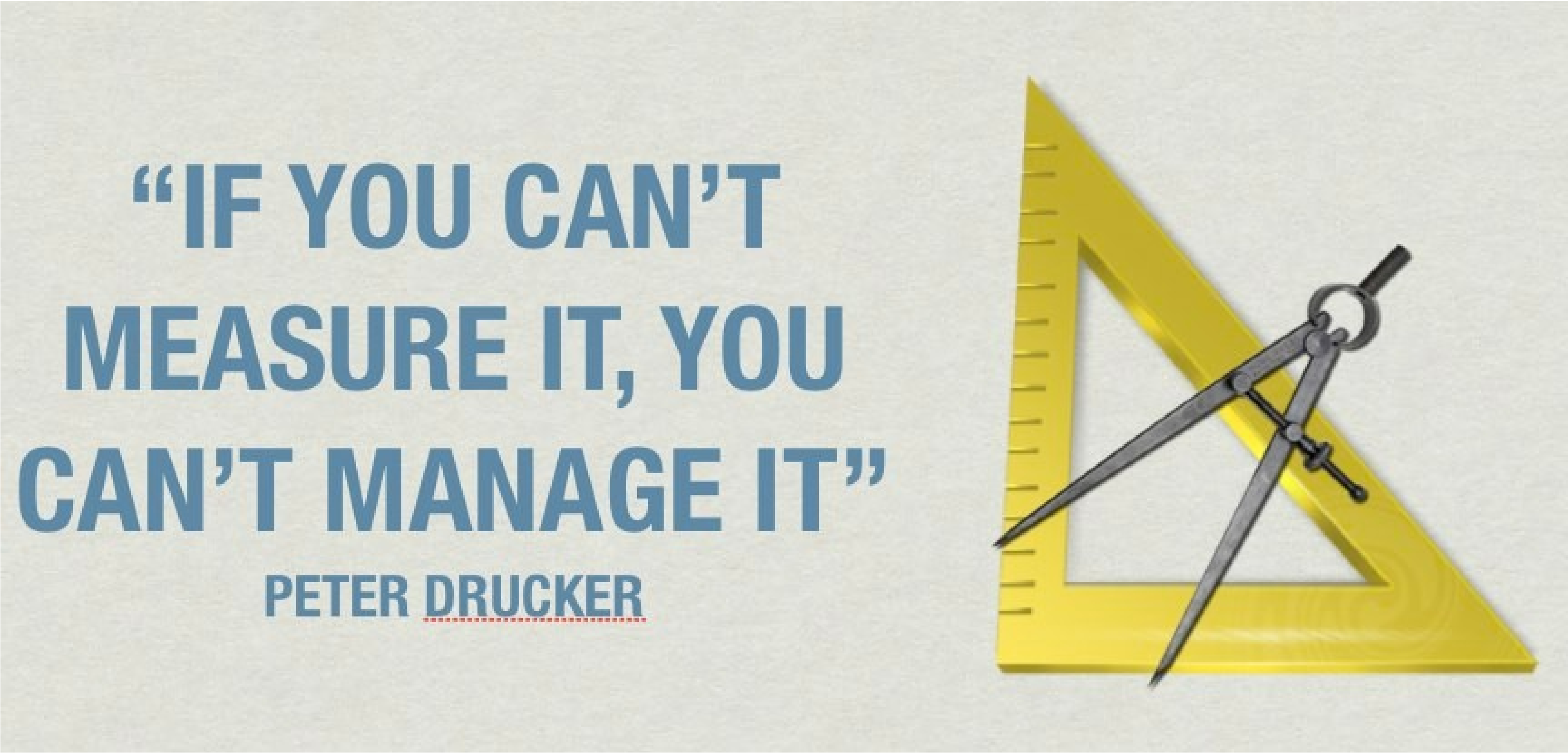
Validation

Testing

Training, Validation, Testing

Joseph Nelson @josephofiowa

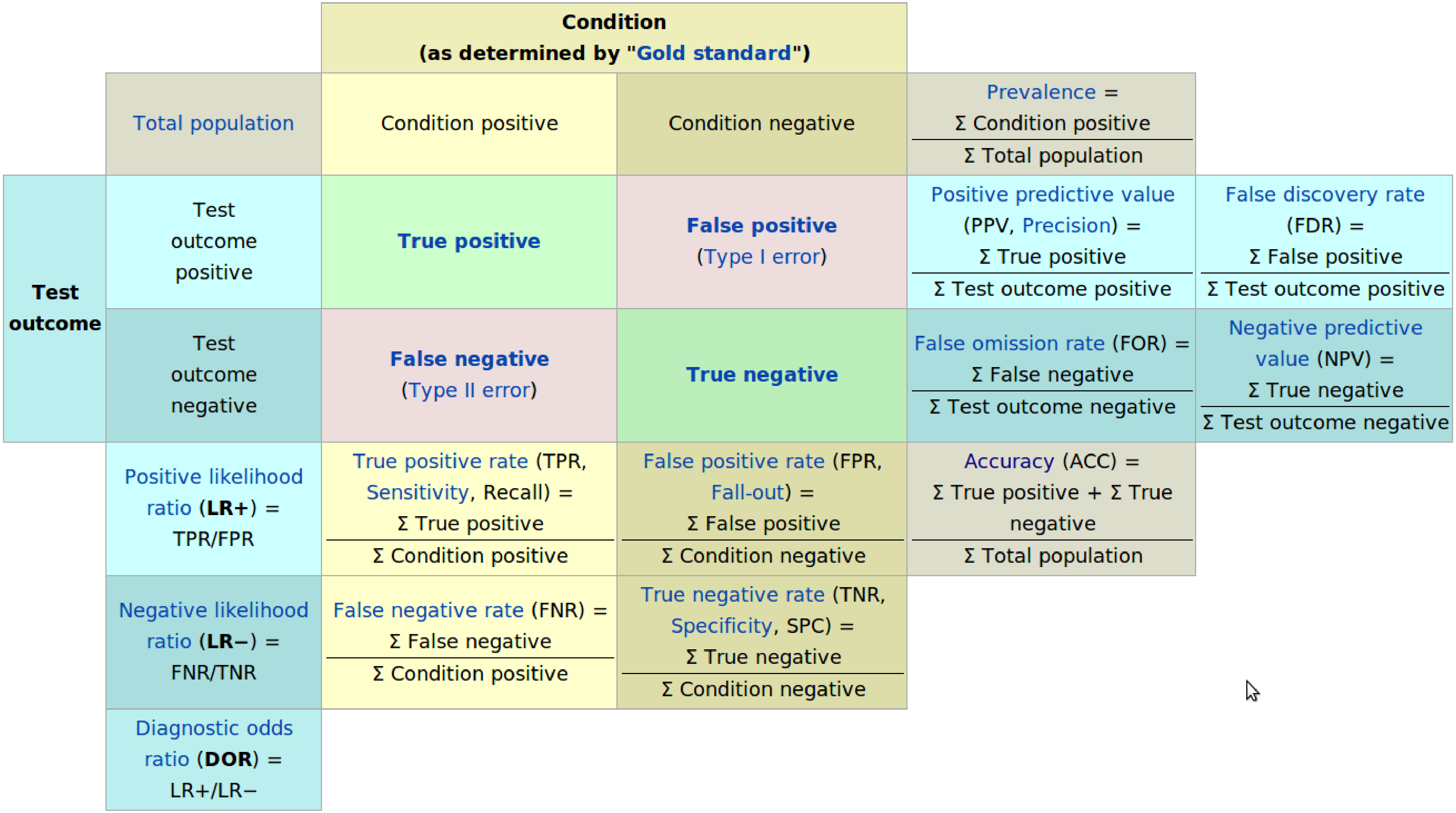
Metrics



"IF YOU CAN'T MEASURE IT, YOU CAN'T MANAGE IT"

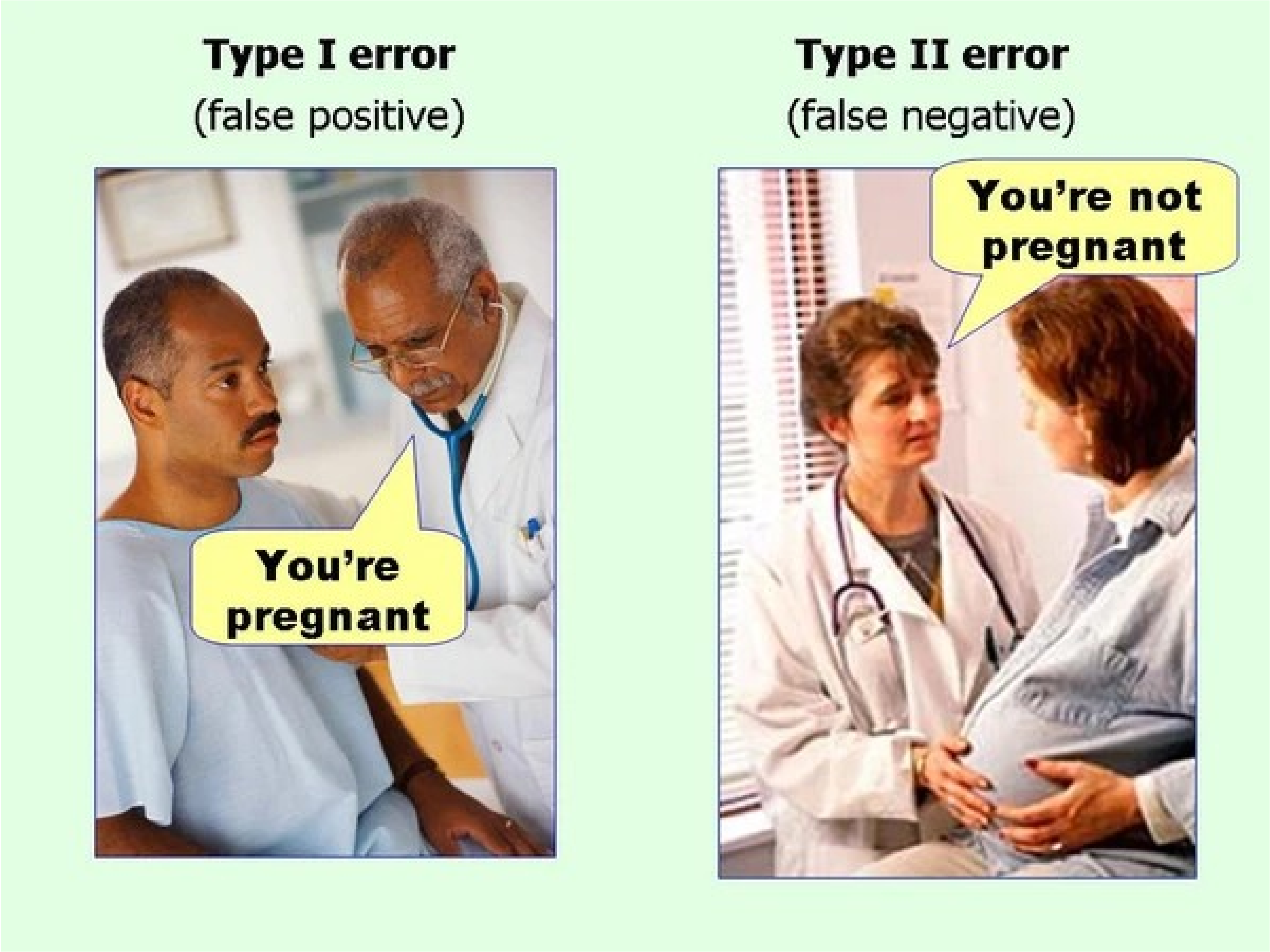
PETER DRUCKER

f1 et al.



		Condition **(as determined by "Gold standard")**			
	Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	
Test outcome	Test outcome positive	**True positive**	**False positive** (Type I error)	Positive predictive value (PPV, Precision) = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Test outcome positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Test outcome positive}}$
	Test outcome negative	**False negative** (Type II error)	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Test outcome negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Test outcome negative}}$
	Positive likelihood ratio (**LR+**) = TPR/FPR	True positive rate (TPR, Sensitivity, Recall) = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR, Fall-out) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
	Negative likelihood ratio (**LR−**) = FNR/TNR	False negative rate (FNR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	True negative rate (TNR, Specificity, SPC) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$		
	Diagnostic odds ratio (**DOR**) = LR+/LR−				

FP vs FN


Type I error
(false positive)
You're pregnant
Type II error
(false negative)
You're not pregnant

f1 et al.

$$F_1 = \frac{2}{\frac{1}{\mathrm{recall}} + \frac{1}{\mathrm{precision}}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{(\beta^2 \cdot \mathrm{precision}) + \mathrm{recall}}$$

$$G = \sqrt{\mathrm{precision} \cdot \mathrm{recall}}$$

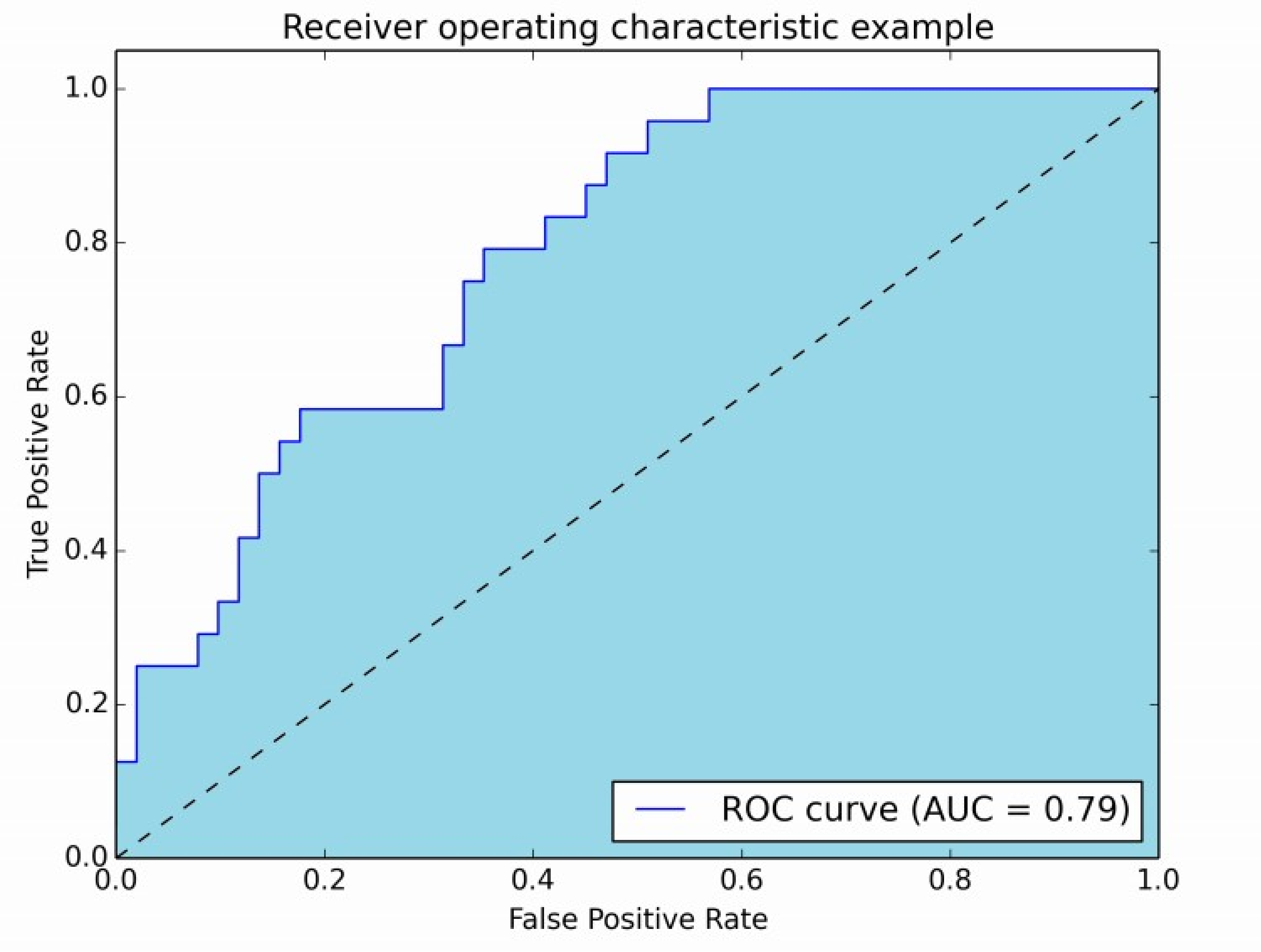
Multilabel f1

* micro - globally count the total true positives, false negatives and false positives

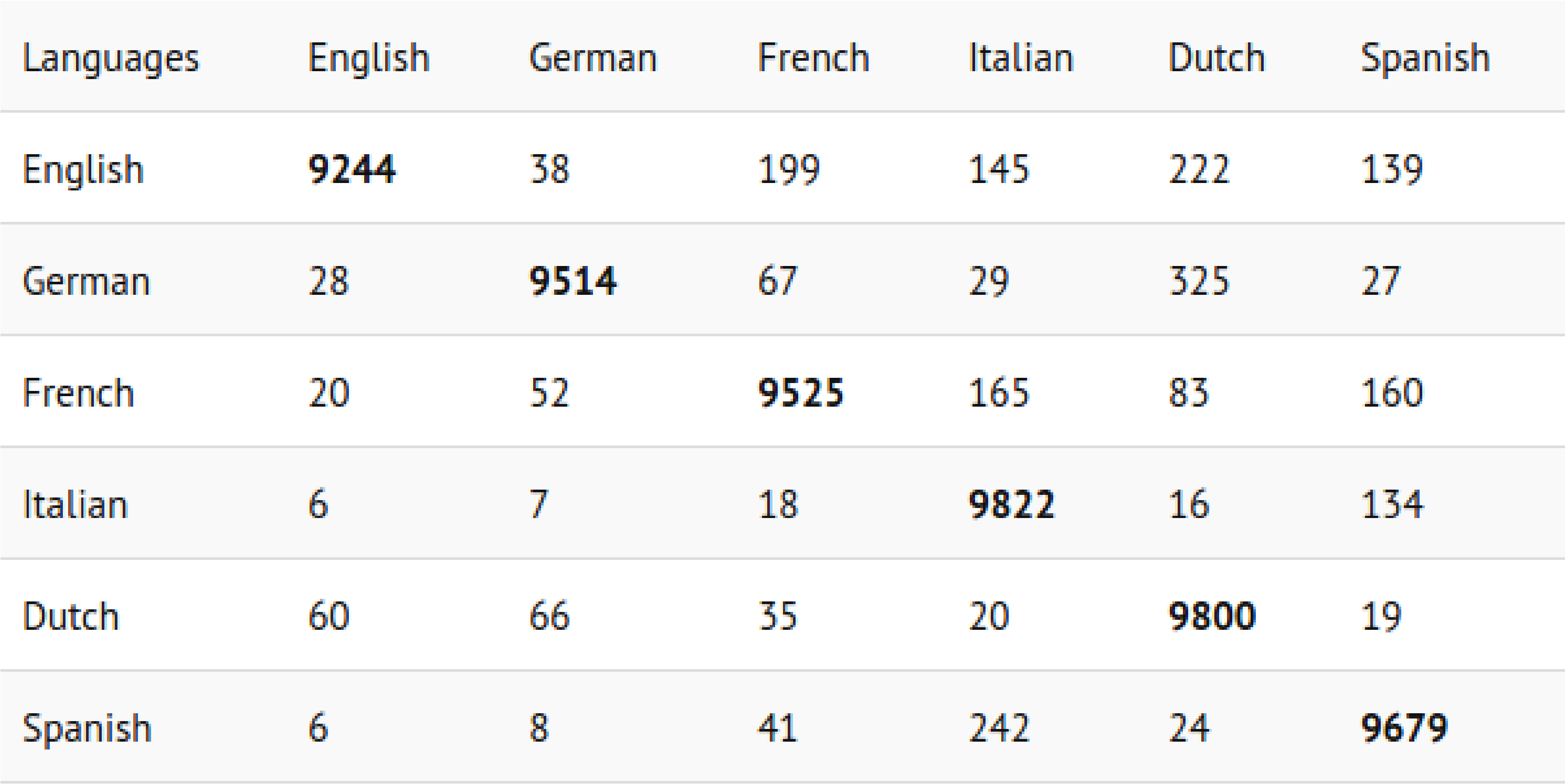
* macro - separately for each label, and take the mean

* weighted - like macro but use weighted mean

ROC AUC


Receiver operating characteristic example
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (AUC = 0.79)

Confusion Matrix



Languages	English	German	French	Italian	Dutch	Spanish
English	**9244**	38	199	145	222	139
German	28	**9514**	67	29	325	27
French	20	52	**9525**	165	83	160
Italian	6	7	18	**9822**	16	134
Dutch	60	66	35	20	**9800**	19
Spanish	6	8	41	242	24	**9679**

Confusion Matrix for WILD

```
AF: 1.00 |
DE: 1.00 |
EN: 1.00 |
ES: 0.94 | IT:0.06
NL: 0.85 | IT:0.03 CA:0.03 AF:0.03 DE:0.03 FR:0.03
RU: 0.82 | UK:0.12 EN:0.03 DA:0.02 FR:0.02
UK: 0.93 | TG:0.07
Total quality: 0.91
```

Training-Related Metrics

1) Cross-entropy

$$H(p, q) = -\sum_{x \in \mathcal{X}} p(x) \, \log q(x)$$

+ KL Divergence (relative entropy)

2) Perplexity

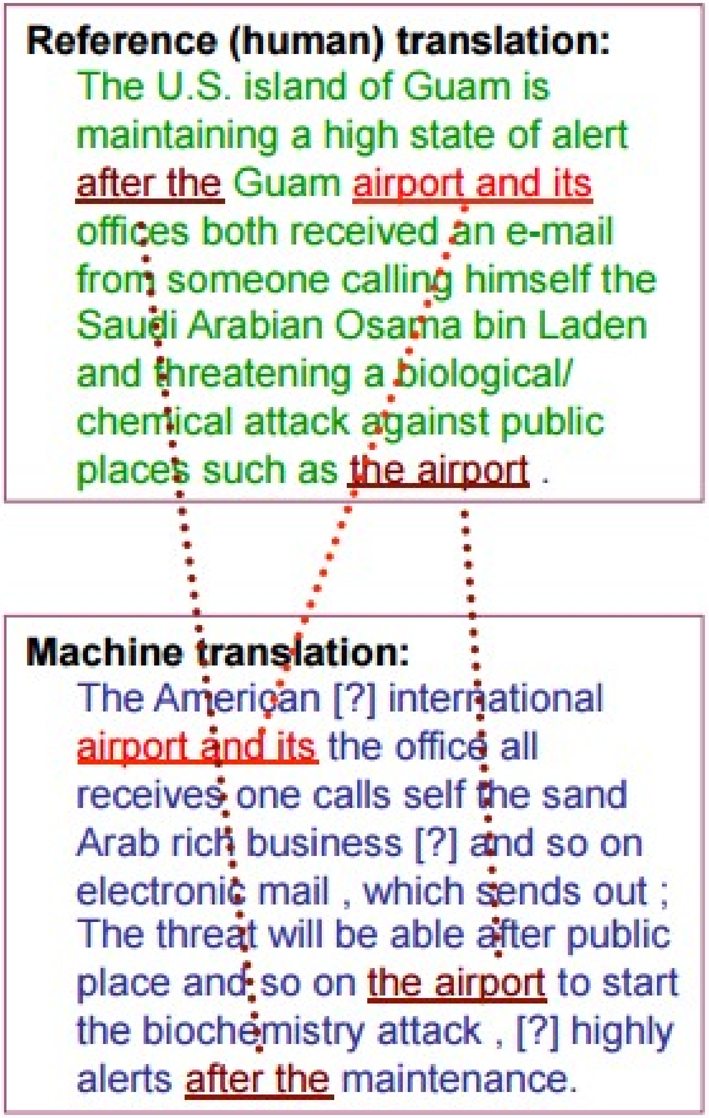
$$2^{H(p)} = 2^{-\sum_x p(x) \log_2 p(x)}$$

Evaluating NLG-like Tasks

1) Word Error Rate (WER)
2) BLEU, ROUGE, METEOR

BLEU Evaluation Metric

(Papineni et al, ACL-2002)



Reference (human) translation:

The U.S. island of Guam is maintaining a high state of alert after the Guam airport and its offices both received an e-mail from someone calling himself the Saudi Arabian Osama bin Laden and threatening a biological/ chemical attack against public places such as the airport .

Machine translation:

The American [?] international airport and its the office all receives one calls self the sand Arab rich business [?] and so on electronic mail , which sends out ; The threat will be able after public place and so on the airport to start the biochemistry attack , [?] highly alerts after the maintenance.

- BLEU is a weighted geometric mean, with a brevity penalty factor added.
  - Note that it's precision-oriented
- BLEU4 formula

  (counts n-grams up to length 4)

$$
\begin{aligned}
\exp (&1.0 * \log p1 + \\
&0.5 * \log p2 + \\
&0.25 * \log p3 + \\
&0.125 * \log p4 - \\
&\max(\text{words-in-reference} / \text{words-in-machine} - 1, 0)
\end{aligned}
$$

p1 = 1-gram precision

P2 = 2-gram precision

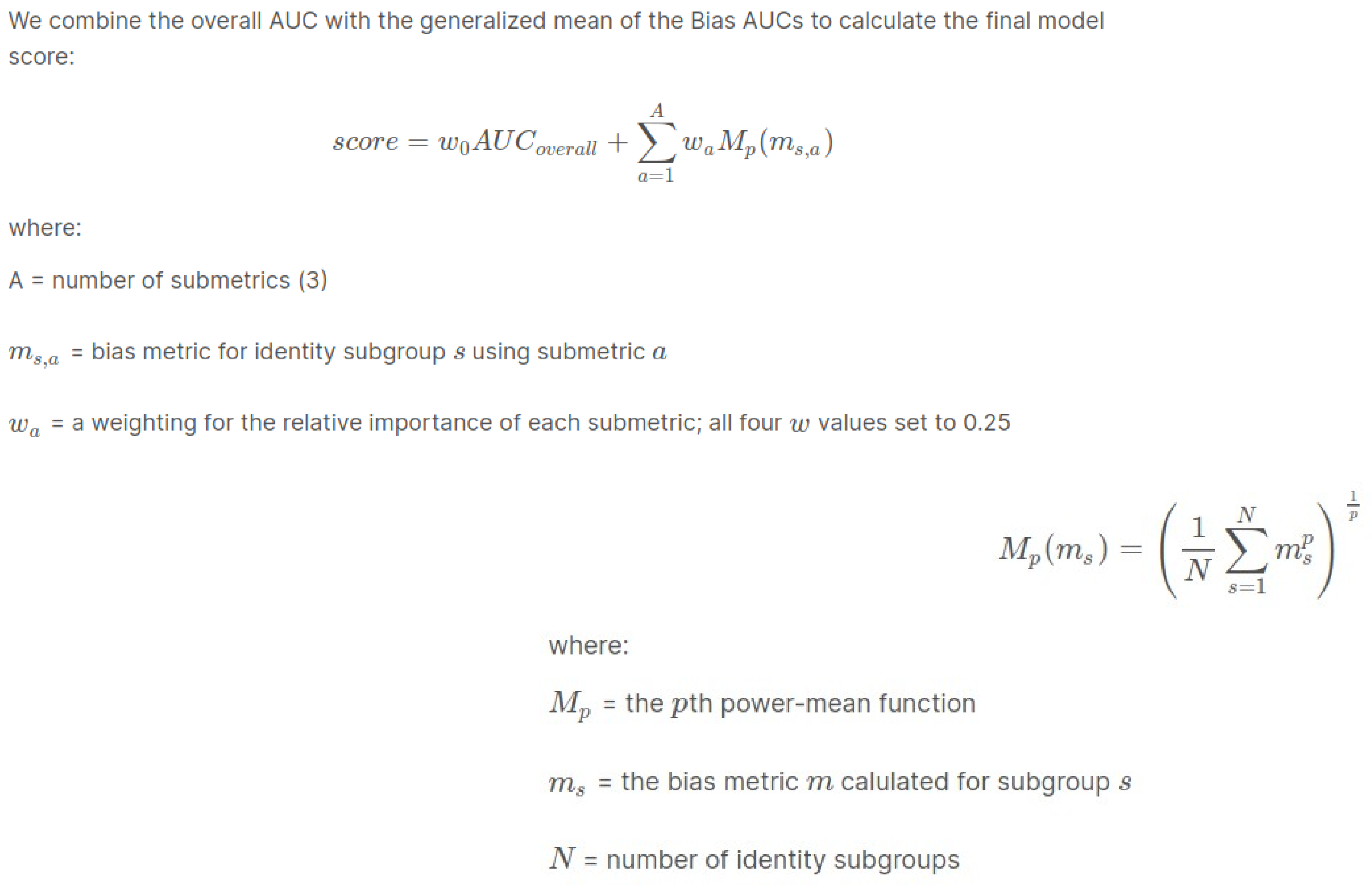
P3 = 3-gram precision

P4 = 4-gram precision

Note: only works at corpus level (zeroes kill it); there's a smoothed variant for sentence-level

Custom Metrics

Jigsaw Toxic comments challenge



We combine the overall AUC with the generalized mean of the Bias AUCs to calculate the final model score:

$$score = w_0 AUC_{overall} + \sum_{a=1}^{A} w_a M_p(m_{s,a})$$

where:

A = number of submetrics (3)

$m_{s,a}$ = bias metric for identity subgroup s using submetric a

w_a = a weighting for the relative importance of each submetric; all four w values set to 0.25

$$M_p(m_s) = \left(\frac{1}{N}\sum_{s=1}^{N} m_s^p\right)^{\frac{1}{p}}$$

where:

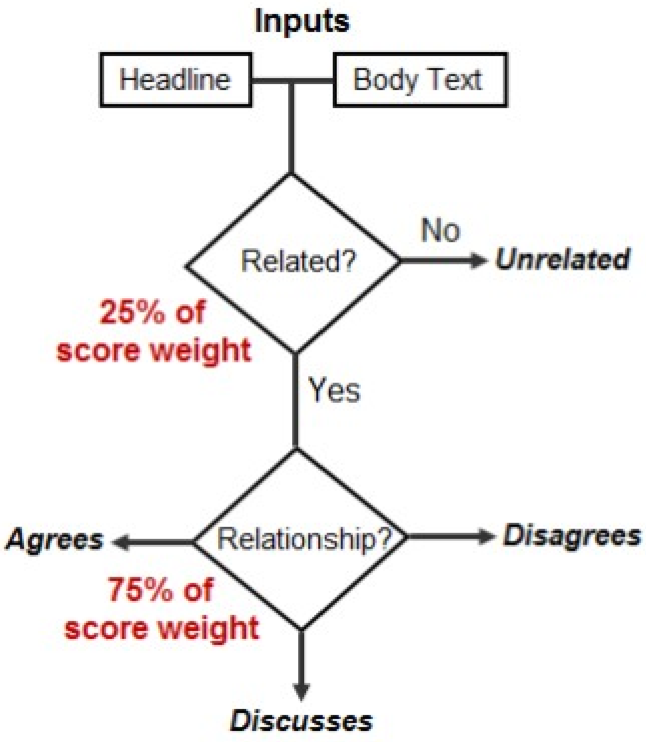
M_p = the pth power-mean function

m_s = the bias metric m calulated for subgroup s

N = number of identity subgroups

Custom Metrics

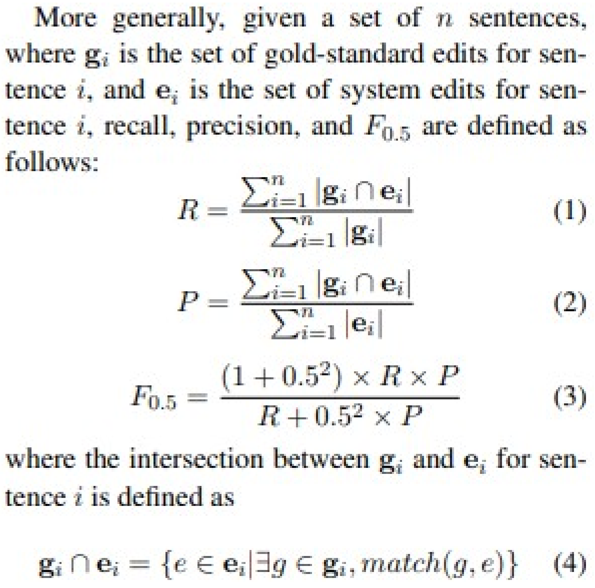
Fake News Challenge (FNC-1)



A simple baseline using hand-coded features and a GradientBoosting classifier is available on github.

Custom Metrics for GEC

- NUS MaxMatch (M^2)



More generally, given a set of n sentences, where $\mathbf{g}_i$ is the set of gold-standard edits for sentence i, and $\mathbf{e}_i$ is the set of system edits for sentence i, recall, precision, and $F_{0.5}$ are defined as follows:

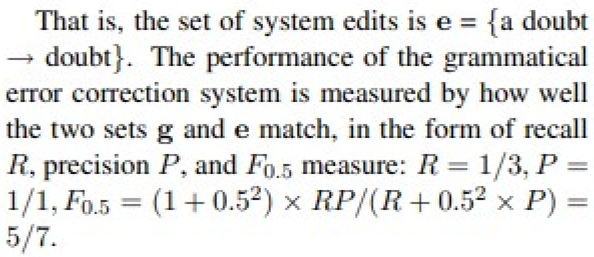
$$R = \frac{\sum_{i=1}^{n} |\mathbf{g}_i \cap \mathbf{e}_i|}{\sum_{i=1}^{n} |\mathbf{g}_i|} \tag{1}$$

$$P = \frac{\sum_{i=1}^{n} |\mathbf{g}_i \cap \mathbf{e}_i|}{\sum_{i=1}^{n} |\mathbf{e}_i|} \tag{2}$$

$$F_{0.5} = \frac{(1 + 0.5^2) \times R \times P}{R + 0.5^2 \times P} \tag{3}$$

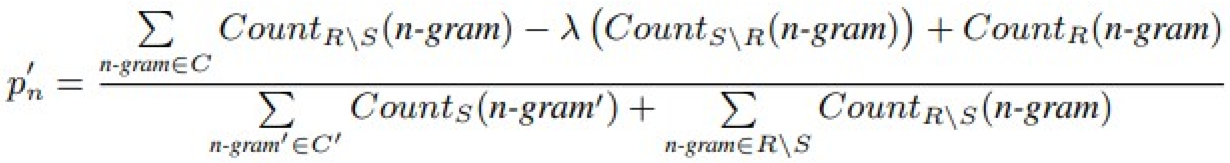
where the intersection between $\mathbf{g}_i$ and $\mathbf{e}_i$ for sentence i is defined as

$$\mathbf{g}_i \cap \mathbf{e}_i = \{e \in \mathbf{e}_i | \exists g \in \mathbf{g}_i, match(g, e)\} \tag{4}$$

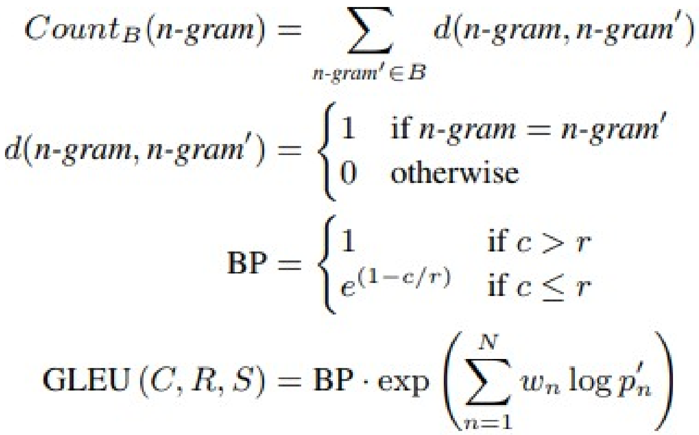


That is, the set of system edits is $\mathbf{e} = \{$a doubt → doubt$\}$. The performance of the grammatical error correction system is measured by how well the two sets $\mathbf{g}$ and $\mathbf{e}$ match, in the form of recall R, precision P, and $F_{0.5}$ measure: $R = 1/3, P = 1/1, F_{0.5} = (1 + 0.5^2) \times RP/(R + 0.5^2 \times P) = 5/7$.

- GLUE



$$p'_n = \frac{\sum_{n\text{-}gram \in C} Count_{R \setminus S}(n\text{-}gram) - \lambda \left(Count_{S \setminus R}(n\text{-}gram) \right) + Count_R(n\text{-}gram)}{\sum_{n\text{-}gram' \in C'} Count_S(n\text{-}gram') + \sum_{n\text{-}gram \in R \setminus S} Count_{R \setminus S}(n\text{-}gram)}$$



$$Count_B(n\text{-}gram) = \sum_{n\text{-}gram' \in B} d(n\text{-}gram, n\text{-}gram')$$

$$d(n\text{-}gram, n\text{-}gram') = \begin{cases} 1 & \text{if } n\text{-}gram = n\text{-}gram' \\ 0 & \text{otherwise} \end{cases}$$

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1 - c/r)} & \text{if } c \leq r \end{cases}$$

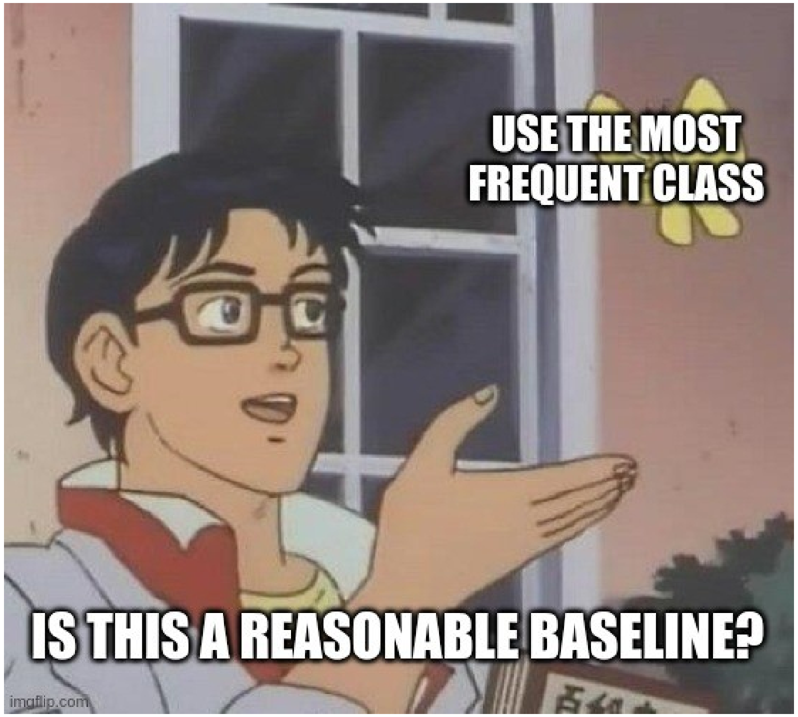
$$\text{GLEU}(C, R, S) = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p'_n \right)$$

- Grammarly Tribunal

https://www.aclweb.org/anthology/P15-2097.pdf

Baseline

Quality that can be achieved using some reasonable primitive approach (on the same data)



2 Views on Quality

* absolute improvement

f1=0.8 => 4 of 5 cases are correctly handled

2 Views on Quality

* absolute improvement

f1=0.8 => 4 of 5 cases are correctly handled

* error reduction

f1 was 0.8, but became 0.82
Is this an improvement?

2 Views on Quality

* absolute improvement

f1=0.8 => 4 of 5 cases are correctly handled

* error reduction

f1 was 0.8, but became 0.82

Is this an improvement?

Only 2% absolute :(

2 Views on Quality

* absolute improvement

f1=0.8 => 4 of 5 cases are correctly handled

* error reduction

f1 was 0.8, but became 0.82
Is this an improvement?
Only 2% absolute :(
But 10% error reduction :)

Recent SNLI Example

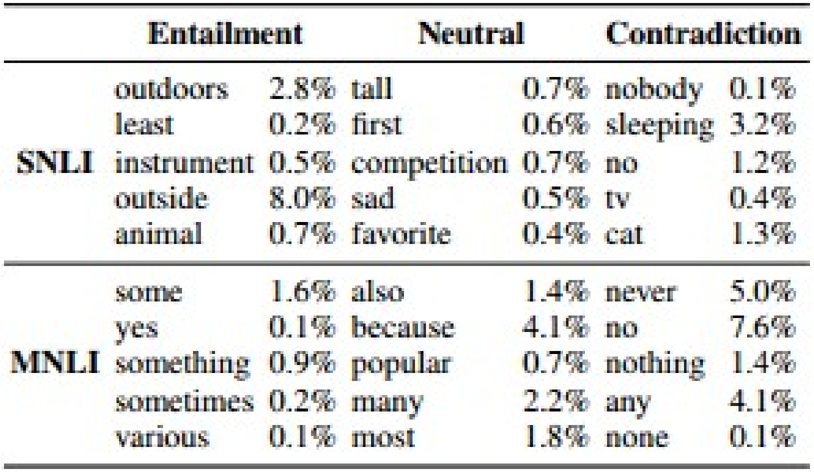
https://arxiv.org/pdf/1803.02324.pdf

- majority class baseline: 0.34
- SOTA models: 0.87

Recent SNLI Example

https://arxiv.org/pdf/1803.02324.pdf

- majority class baseline: 0.34
- SOTA models: 0.87
- basic fasttext classifier: 0.67



	Entailment		Neutral		Contradiction	
SNLI	outdoors	2.8%	tall	0.7%	nobody	0.1%
	least	0.2%	first	0.6%	sleeping	3.2%
	instrument	0.5%	competition	0.7%	no	1.2%
	outside	8.0%	sad	0.5%	tv	0.4%
	animal	0.7%	favorite	0.4%	cat	1.3%
MNLI	some	1.6%	also	1.4%	never	5.0%
	yes	0.1%	because	4.1%	no	7.6%
	something	0.9%	popular	0.7%	nothing	1.4%
	sometimes	0.2%	many	2.2%	any	4.1%
	various	0.1%	most	1.8%	none	0.1%

Table 4: Top 5 words by PMI(*word*, *class*), along with the proportion of *class* training samples containing *word*. MultiNLI is abbreviated to MNLI.

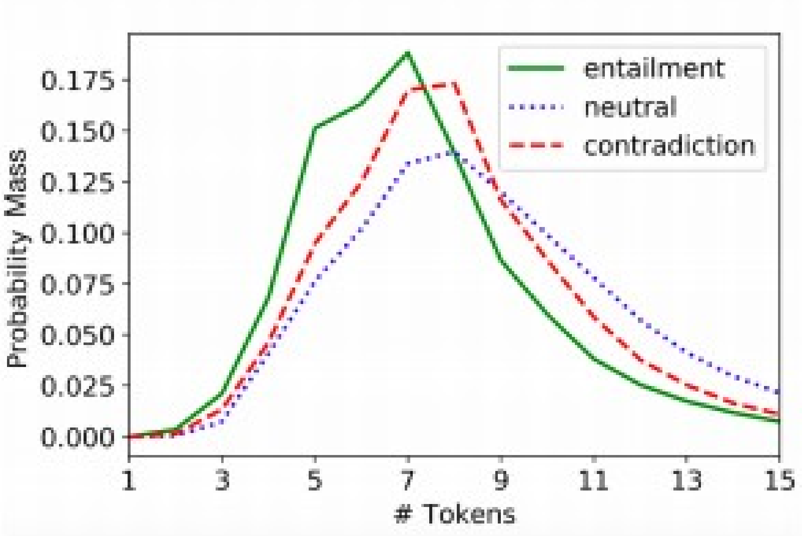


Figure 1: The probability mass function of the hypothesis length in SNLI, by class.

State-of-the-Art (SOTA)

The highest publicly known result on a well-established dataset

https://aclweb.org/aclwiki/State_of_the_art

GLUE Benchmark

The General Language Understanding Evaluation (GLUE) benchmark is a collection of resources for training, evaluating, and analyzing natural language understanding systems. GLUE consists of:

* A benchmark of nine sentence- or sentence-pair language understanding tasks built on established existing datasets and selected to cover a diverse range of dataset sizes, text genres, and degrees of difficulty
* A diagnostic dataset designed to evaluate and analyze model performance with respect to a wide range of linguistic phenomena found in natural language, and
* A public leaderboard for tracking performance on the benchmark and a dashboard for visualizing the performance of models on the diagnostic set.

decaNLP

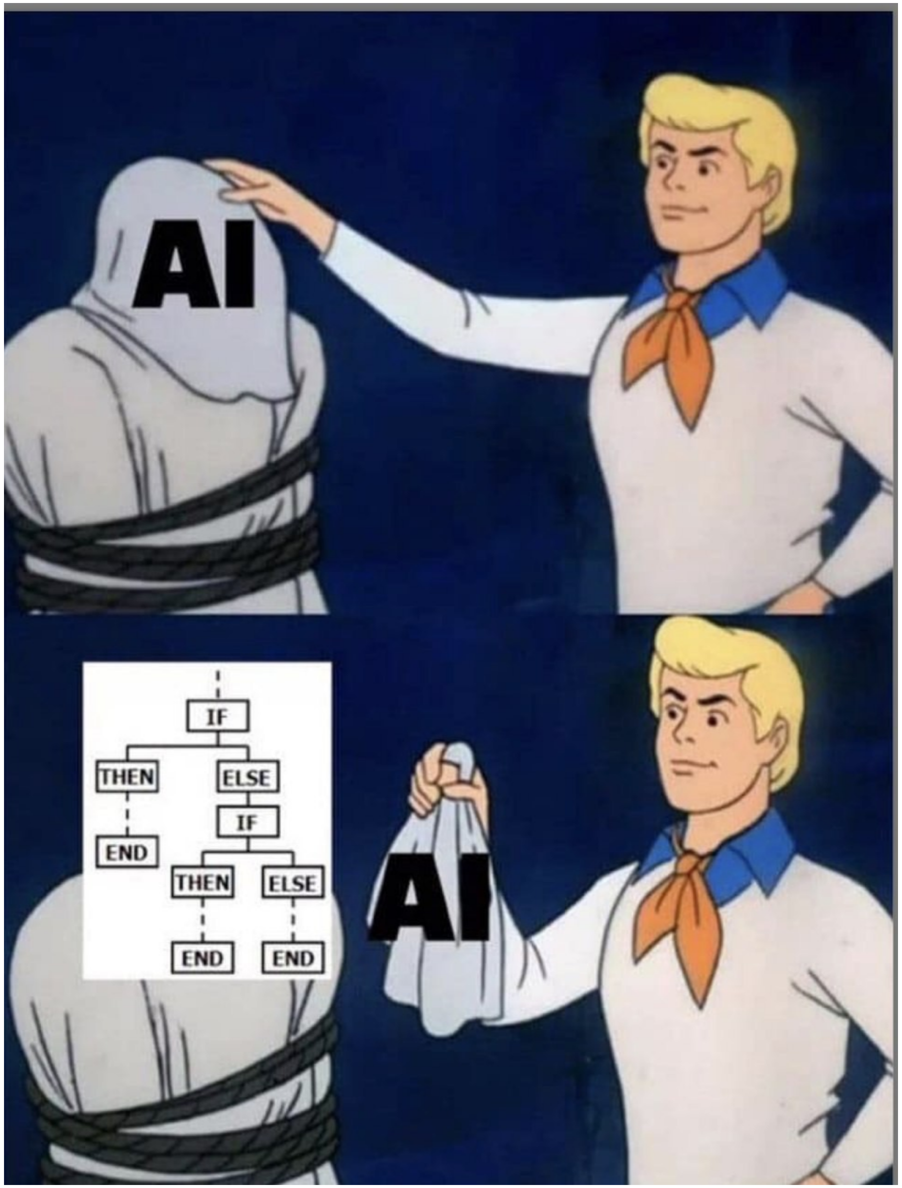
The Natural Language Decathlon is a multitask challenge that spans ten tasks: question answering (SQuAD), machine translation (IWSLT), summarization (CNN/DM), natural language inference (MNLI), sentiment analysis (SST), semantic role labeling(QA-SRL), zero-shot relation extraction (QA-ZRE), goal-oriented dialogue (WOZ, semantic parsing (WikiSQL), and commonsense reasoning (MWSC). Each task is cast as question answering, which makes it possible to use our new Multitask Question Answering Network (MQAN).

Evolution of NLProc Paradigms

1. Symbolical/rule-based
   “The (Weak) Empire of Reason”
2. Counting-based
   “The Empiricist Invasion”
3. Deep Learning-based
   “The Revenge of the Spherical Cows”

http://www.earningmyturns.org/2017/06/a-computational-linguistic-farce-in.html

Rule-based Approach



English Stemming
(rule-based definition)

Porter stemmer:

```
(defun stem (word)
  (if (<= (length word) 2)
      word
      (-> word step1ab step1c step2 step3 step4 step5)))

step1ab: (-> word stem-s stem-ed/ing)
stem-s:  (when (and (ends-with ”s” word)
               (not (ends-with ”ss” word)))
           (substr word (if (or (ends-with ”sses” word)
                                (ends-with ”ies” word))
                            -2 -1)))
step1c:  (when (and (ends-with ”y” word)
               (vowel-in-stem? word 1))
           (:= (last-char word) #\i))
...
```

https://github.com/vseloved/cl-nlp/blob/master/src/lexics/porter.lisp

English Tokenization
(small problem)

```
* Simplest regex:
  [^\s]+
* More advanced regex:
  \w+|[!"#$%&'*+,\./:;<=>?@^`~…\(\)⟨⟩{}\[\|\]‒–—―«»“”‘’-]
* Even more advanced regex:
  [+-]?[0-9](?:[0-9,\.]*[0-9])?
  |[\w@](?:[\w'’`@-][\w']|[\w'][\w@'’`-])*[\w']?
  |["#$%&*+,/:;<=>@^`~…\(\)⟨⟩{}\[\|\]‒–—―«»“”‘’']
  |[\.!?]+
  |-+

Add post-processing:
* concatenate abbreviations and decimals
* split contractions with regexes
  - 2-character abbreviations regex:
    i['‘’`]m|(?:s?he|it)['‘’`]s|(?:i|you|s?he|we|they)['‘’`]d$
  - 3-character abbreviations regex:
    (?:i|you|s?he|we|they)['‘’`](?:ll|[vr]e)|n['‘’`]t$
```

https://github.com/lang-uk/tokenize-uk/blob/master/tokenize_uk/tokenize_uk.py

Error Correction
(inherently rule-based)

LanguageTool:

```
<category name="Стиль" id="STYLE" type="style">
    <rulegroup id="BILSH_WITH_ADJ" name="Більш з прикметниками">
      <rule>
        <pattern>
          <token>більш</token>
          <token postag_regexp="yes" postag="ad[jv]:.*compc.*">
            <exception>менш</exception>
          </token>
        </pattern>
        <message>Після «більш» не може стояти вища форма прикметника</message>
        <!-- TODO: when we can bind comparative forms togher again
            <suggestion><match no="2"/></suggestion>
            <suggestion><match no="1"/> <match no="2" postag_regexp="yes"
postag="(.*):compc(.*)" postag_replace="$1:compb$2"/></suggestion>
            <example correction="Світліший|Більш світлий"><marker>Більш
світліший</marker>.</example>
        -->
        <example correction=""><marker>Більш світліший</marker>.</example>
        <example>все закінчилось більш менш</example>
      </rule>
...
```

https://github.com/languagetool-org/languagetool/blob/master/languagetool-language-modules/uk/src/main/resources/org/languagetool/rules/uk/

Spelling Correction
(simple statistic-based)

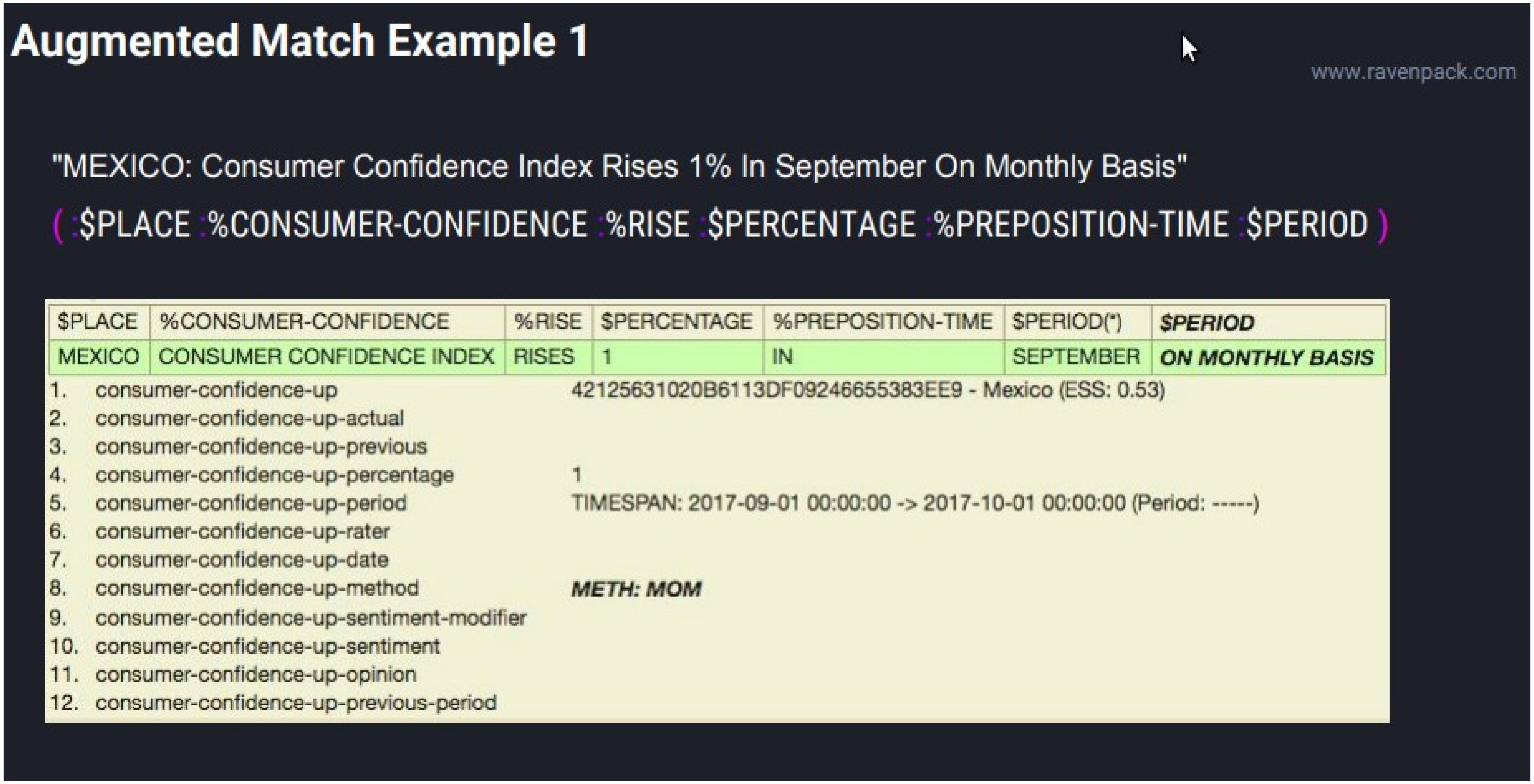
Noisy Channel Model:

```
<category name="Стиль" id="STYLE" type="style">
    <rulegroup id="BILSH_WITH_ADJ" name="Більш з прикметниками">
      <rule>
        <pattern>
          <token>більш</token>
          <token postag_regexp="yes" postag="ad[jv]:.*compc.*">
            <exception>менш</exception>
          </token>
        </pattern>
        <message>Після «більш» не може стояти вища форма прикметника</message>
        <!-- TODO: when we can bind comparative forms togher again
            <suggestion><match no="2"/></suggestion>
            <suggestion><match no="1"/> <match no="2" postag_regexp="yes"
postag="(.*):compc(.*)" postag_replace="$1:compb$2"/></suggestion>
            <example correction="Світліший|Більш світлий"><marker>Більш
світліший</marker>.</example>
        -->
        <example correction=""><marker>Більш світліший</marker>.</example>
        <example>все закінчилось більш менш</example>
      </rule>
...
```

https://github.com/languagetool-org/languagetool/blob/master/languagetool-language-modules/uk/src/main/resources/org/languagetool/rules/uk/

Information Extraction
(suited for iterative approach)

Ravenpack news analytics:



Augmented Match Example 1

www.ravenpack.com

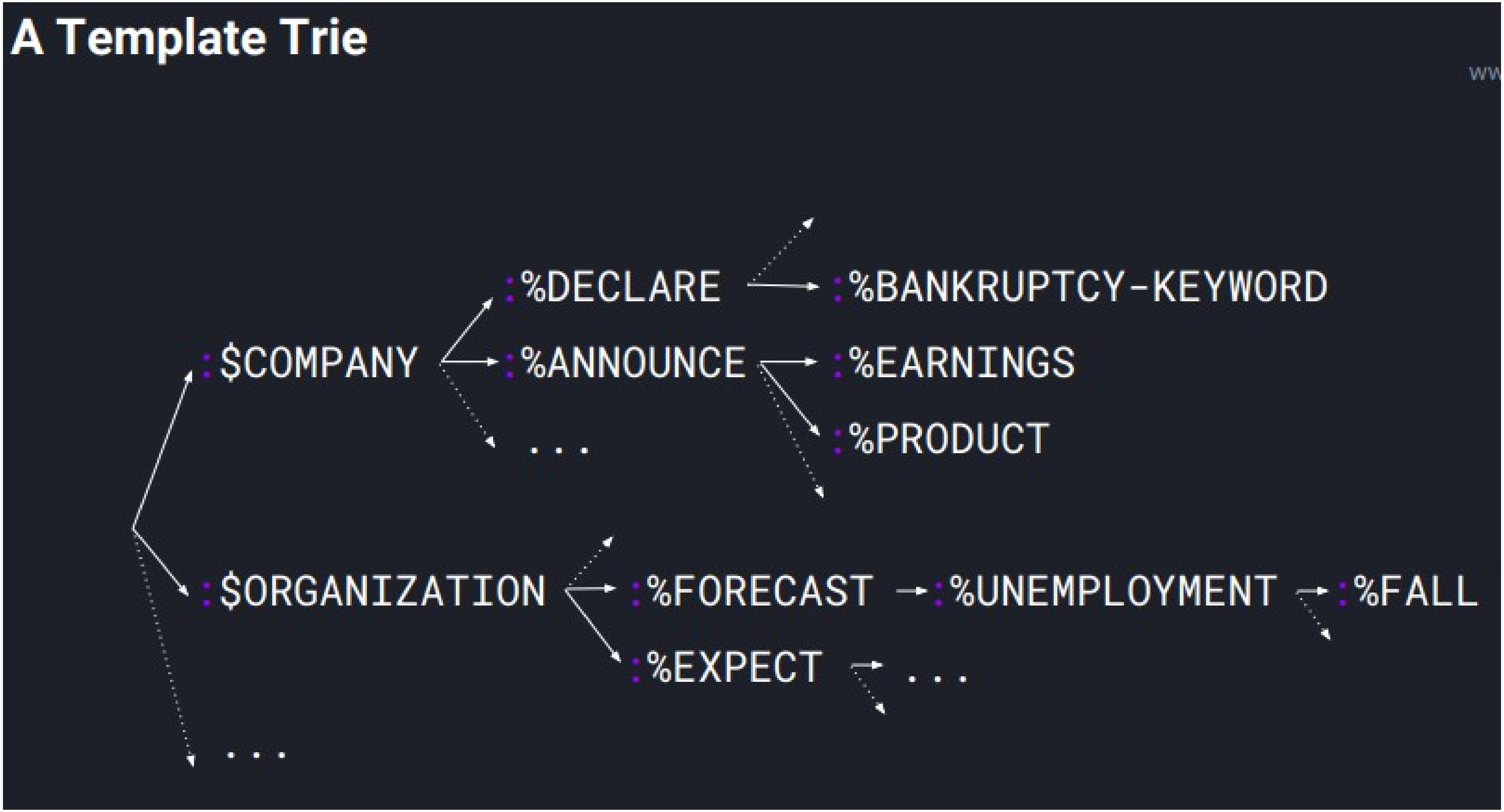
"MEXICO: Consumer Confidence Index Rises 1% In September On Monthly Basis"

(:$PLACE :%CONSUMER-CONFIDENCE :%RISE :$PERCENTAGE :%PREPOSITION-TIME :$PERIOD)

$PLACE	%CONSUMER-CONFIDENCE	%RISE	$PERCENTAGE	%PREPOSITION-TIME	$PERIOD(*)	***$PERIOD***
MEXICO	CONSUMER CONFIDENCE INDEX	RISES	1	IN	SEPTEMBER	***ON MONTHLY BASIS***

1.	consumer-confidence-up	42125631020B6113DF09246655383EE9 - Mexico (ESS: 0.53)
2.	consumer-confidence-up-actual	
3.	consumer-confidence-up-previous	
4.	consumer-confidence-up-percentage	1
5.	consumer-confidence-up-period	TIMESPAN: 2017-09-01 00:00:00 -> 2017-10-01 00:00:00 (Period: -----)
6.	consumer-confidence-up-rater	
7.	consumer-confidence-up-date	
8.	consumer-confidence-up-method	***METH: MOM***
9.	consumer-confidence-up-sentiment-modifier	
10.	consumer-confidence-up-sentiment	
11.	consumer-confidence-up-opinion	
12.	consumer-confidence-up-previous-period	

https://european-lisp-symposium.org/static/2018/cornez.pdf



A Template Trie

:$COMPANY

:%DECLARE

:%BANKRUPTCY-KEYWORD

:%ANNOUNCE

:%EARNINGS

:%PRODUCT

...

:$ORGANIZATION

:%FORECAST

:%UNEMPLOYMENT

:%FALL

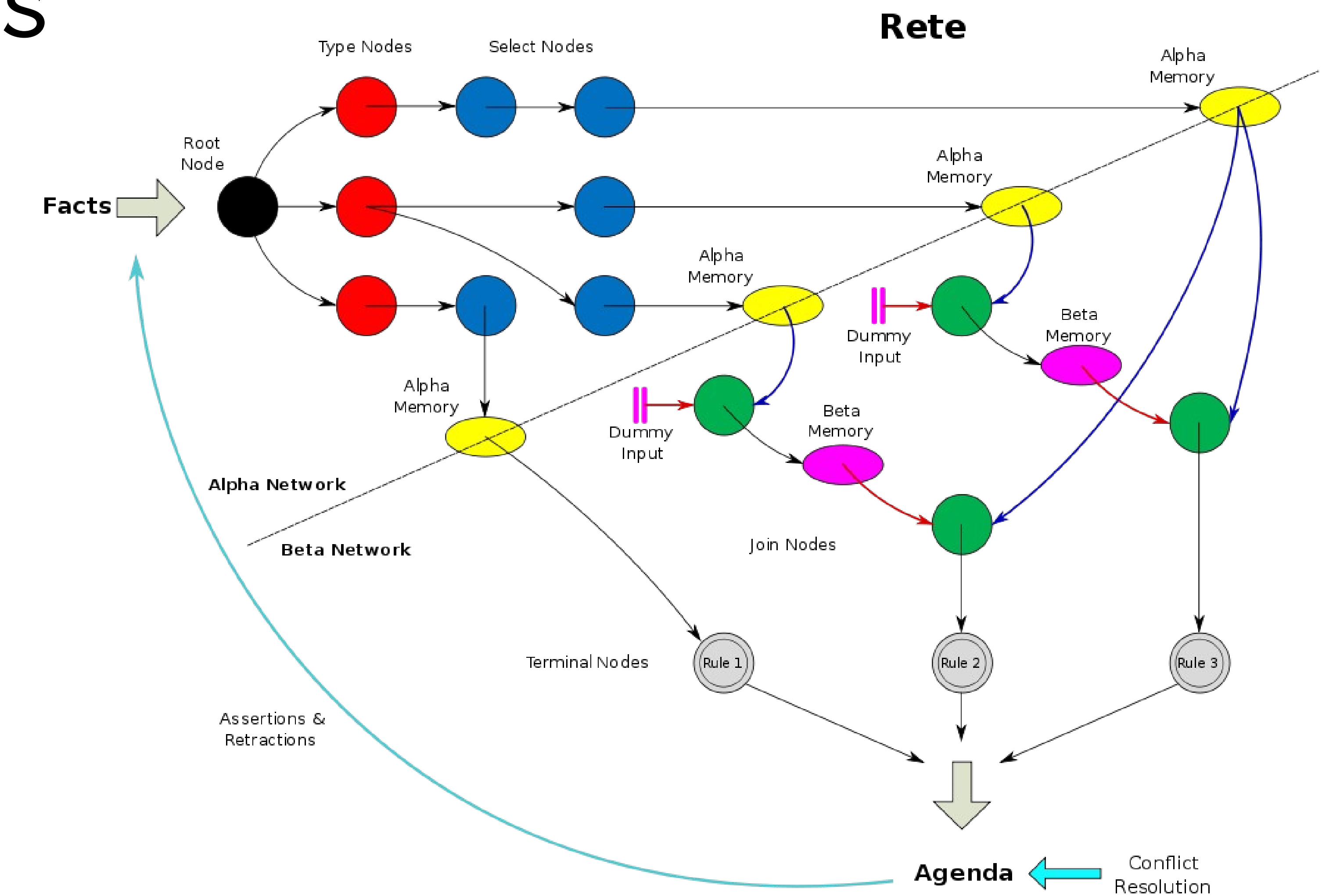
:%EXPECT

...

...

Rete Algorithm

Efficient matching of many templates



Dialog Systems
(inherently scripted)

ELIZA:

```
(defparameter *eliza-rules*
 '((((?* ?x) hello (?* ?y))
    (How do you do.  Please state your problem.))
   (((?* ?x) I want (?* ?y))
    (What would it mean if you got ?y)
    (Why do you want ?y) (Suppose you got ?y soon))
   (((?* ?x) if (?* ?y))
    (Do you really think its likely that ?y) (Do you wish that ?y)
    (What do you think about ?y) (Really-- if ?y))
   (((?* ?x) no (?* ?y))
    (Why not?) (You are being a bit negative)
    (Are you saying "NO" just to be negative?))
   (((?* ?x) I was (?* ?y))
    (Were you really?) (Perhaps I already knew you were ?y)
    (Why do you tell me you were ?y now?))
   (((?* ?x) I feel (?* ?y))
    (Do you often feel ?y ?))
   (((?* ?x) I felt (?* ?y))
    (What other feelings do you have?))))
```

https://norvig.com/paip/eliza1.lisp

(…and if you thought rules are dead)

Wit.ai:

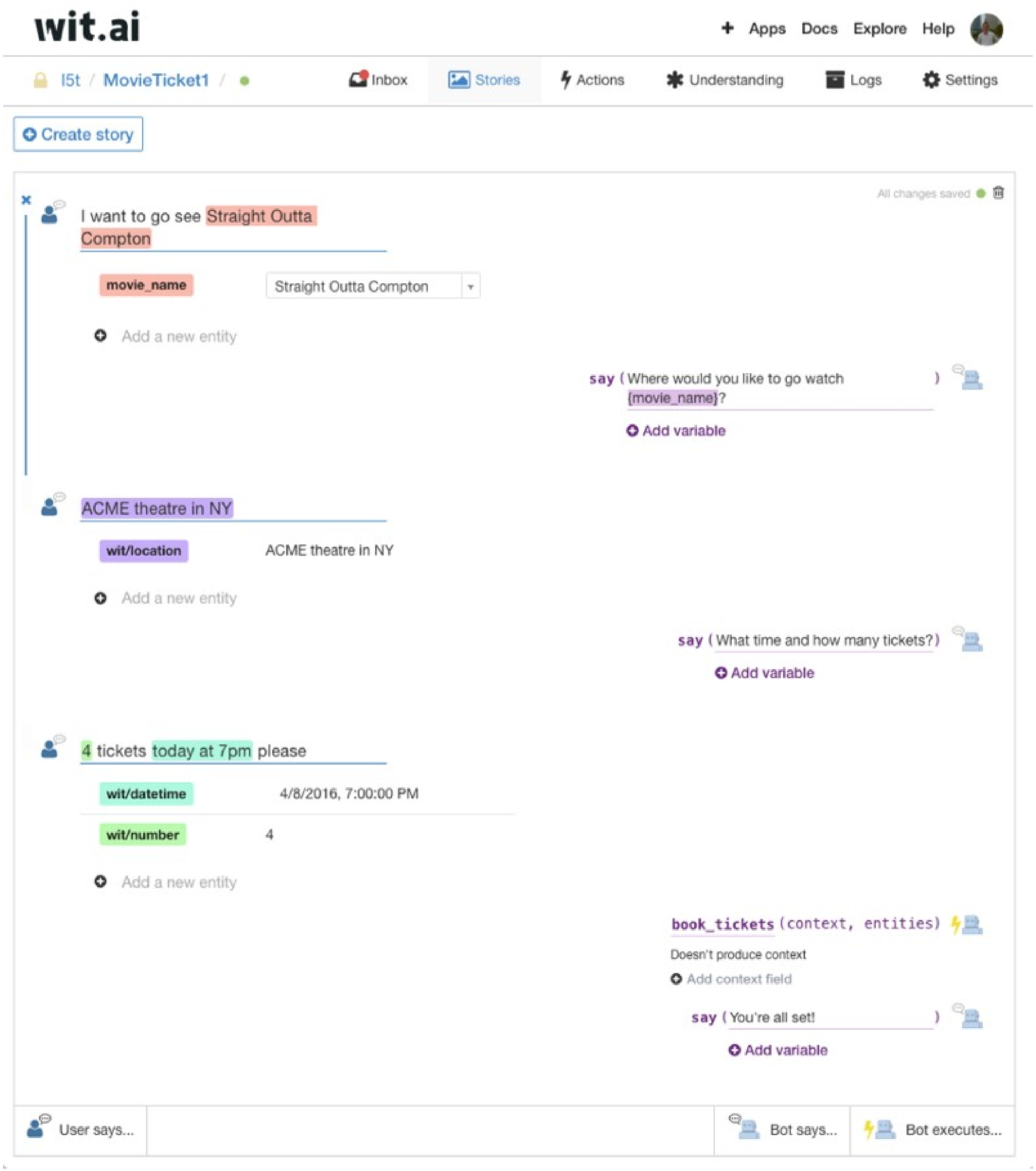
To make a bot, there are two schools: rules or machine learning. (Everybody claims rules are bad and their bot is powered by AI, but when you really look under the hood, the core is often imperative.)

Machine learning is of course more desirable, but the problem is the training dataset. Training a Wit intent with a dozen examples works well, and it’s easy to leverage the community to get more examples. But in order to entirely learn the business logic of a bot of medium complexity, we would need many, many thousands of example conversations.

Rules (or any kind of imperative approach, including plain script/program) are kind of the opposite. The good thing with rules is, you can have a demo working after you write two rules. As long as you follow the script carefully, your bot will work and your audience will be impressed. But as you discover new “paths” in the dialog, you’ll add more and more rules, until one day everything collapses. You’re doomed by the curse of combinatorics. Any new rule conflicts with old rules that you totally forgot the reason for. Your bot cannot improve anymore.

When you create your bot, you just start with a few stories that describe the most probable conversations paths. At this stage, Bot Engine will build a machine learning model that deliberately overfits the stories dataset. Practically, it means that stories will behave almost like rules.

https://medium.com/wit-ai/bot-engine-26af22d37fd6



I want to go see Straight Outta Compton

movie_name	Straight Outta Compton

say (Where would you like to go watch {movie_name}?)

ACME theatre in NY

wit/location	ACME theatre in NY

say (What time and how many tickets?)

4 tickets today at 7pm please

wit/datetime	4/8/2016, 7:00:00 PM
wit/number	4

```
book_tickets (context, entities)
```

Doesn't produce context

say (You're all set!)

Fighting Spam
(historically bad idea)

SpamAssasin:

```
# bodyn
# example: postacie greet! Chcesz porozmawiac  Co prawda tu rzadko bywam,
zwykle pisze tu -  http://hanna.3xa.info
uri      __LOCAL_LINK_INFO  /http:\/\/\w{3,8}\.\w\w\w\.info/
header   __LOCAL_FROM_O2    From =~ /\@o2\.pl/
meta     LOCAL_LINK_INFO    __LOCAL_LINK_INFO && __LOCAL_FROM_O2
describe LOCAL_LINK_INFO    Link postaci http://cos.cos.info i z o2.pl
score    LOCAL_LINK_INFO    5
```

https://wiki.apache.org/spamassassin/CustomRulesets

(…which continues being reimplemented)

Facebook antispam (using HAXL):

```
fpSpammer :: Haxl Bool
fpSpammer =
  talkingAboutFP .&&
  numFriends .> 100 .&&
  friendsLikeCPlusPlus
 where
  talkingAboutFP =
    strContains “Functional Programming” <$> postContent

friendsLikeCPlusPlus = do
  friends <- getFriends
  cppFriends <- filterM likesCPlusPlus friends
  return (length cppFriends >= length friends `div` 2)
```

http://multicore.doc.ic.ac.uk/iPr0gram/slides/2015-2016/Marlow-fighting-spam.pdf

https://petrimazepa.com/m_li_ili_kak_algoritm_facebook_blokiruet_polzovatelei_na_osnovanii_stop_slov

More Rule-based Examples

* Rule-based MT (Systran)
* Various NLG systems
* Custom parsing (Sparser, Zoral)

Languages for Rules

* regexes

```
[+-]?[0-9](?:[0-9,\.]*[0-9])?
|[\w@](?:[\w’'`@-][\w’]|[\w’][\w@’'`-])*[\w’]?
|[”#$%&*+,/:;<=>@^`~…\(\)⟨⟩{}\[\|\]———«»“”‘’']
|[\.!?]+
|-+
```

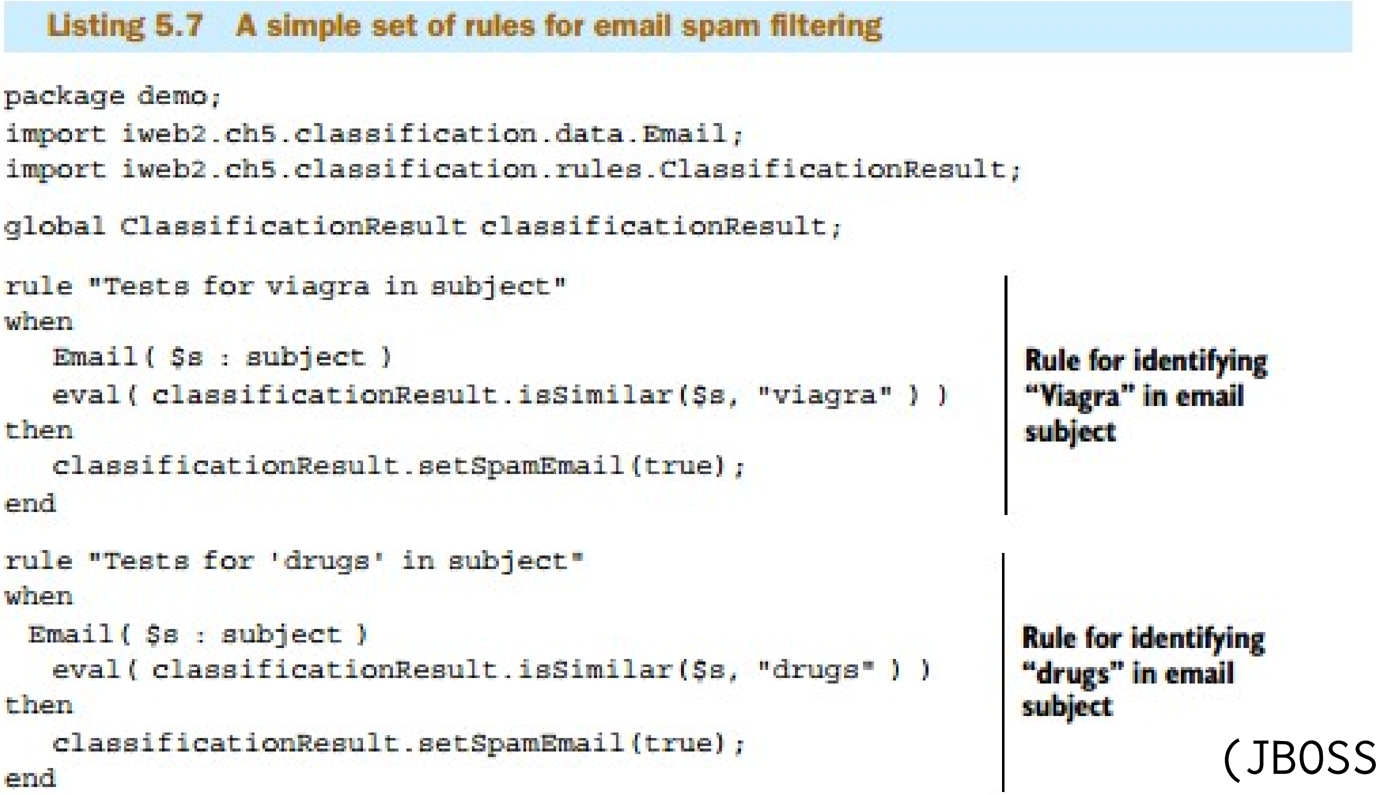
Languages for Rules

* regexes
* XML, JSON

```
<category name="Стиль" id="STYLE" type="style">
    <rulegroup id="BILSH_WITH_ADJ" name="Більш з прикметниками">
      <rule>
        <pattern>
          <token>більш</token>
          <token postag_regexp="yes" postag="ad[jv]:.*compc.*">
            <exception>менш</exception>
          </token>
        </pattern>
        <message>Після «більш» не може стояти вища форма прикметника</message>
        <!-- TODO: when we can bind comparative forms togher again
            <suggestion><match no="2"/></suggestion>
```

Languages for Rules

- * regexes
- * XML, JSON
- * external DSLs



Listing 5.7 A simple set of rules for email spam filtering

```
package demo;
import iweb2.ch5.classification.data.Email;
import iweb2.ch5.classification.rules.ClassificationResult;

global ClassificationResult classificationResult;

rule "Tests for viagra in subject"
when
   Email( $s : subject )
   eval( classificationResult.isSimilar($s, "viagra" ) )
then
   classificationResult.setSpamEmail(true);
end

rule "Tests for 'drugs' in subject"
when
  Email( $s : subject )
   eval( classificationResult.isSimilar($s, "drugs" ) )
then
   classificationResult.setSpamEmail(true);
end
```

Rule for identifying "Viagra" in email subject

Rule for identifying "drugs" in email subject

(JBOSS drooles)

Languages for Rules

* regexes
* XML, JSON
* external DSLs
* internal DSLs

```
(match-html source
            '(>> article
                 (aside (>> a ($ user))
                        (>> li (strong ”Native Tongue:”) ($ lang)))
                 (div |...| (>> (div :data-role ”commentContent”
                                     ($ text) (span) |...|))
                      !!!))
```

https://github.com/vseloved/crawlik

Rules Pros&Cons

+ compact & fast
+ full control
+ arbitrary recall
+ iterative
+ best interpetability
+ perfectly accommodate domain experts

- precision ceiling
- non-optimal weights
- require a lot of human labor
- hard to interpret/calibrate score

Hybrid Approach

“Rule-based” framework that incorporates Machine Learning models.

+ control
+ best of both worlds
- more complex
- not sexy

Hybrid WILD

Rule-based framework incorporating ML

1. Tokenize
2. Filter non-words
3. Process each word:
   a. Determine script
   b. For single-lang scripts — output lang
   c. For multi-lang scripts — run classifier for the language selection
      - whole-word classifier
      - or 3gram classifier
4. Combine per-word result weighted by word length
5. Weight by the supplied language distribution

More Uses of Rules

* getting data
* pre- and post-processing
* feature extraction
* data generation
* prototyping

Counteracting Bias

* Rule-based post-processing
* Mix multiple datasets
  + special feature-selection
* Train multiple models on different datasets and create an ensemble
* Use domain adaptation techniques

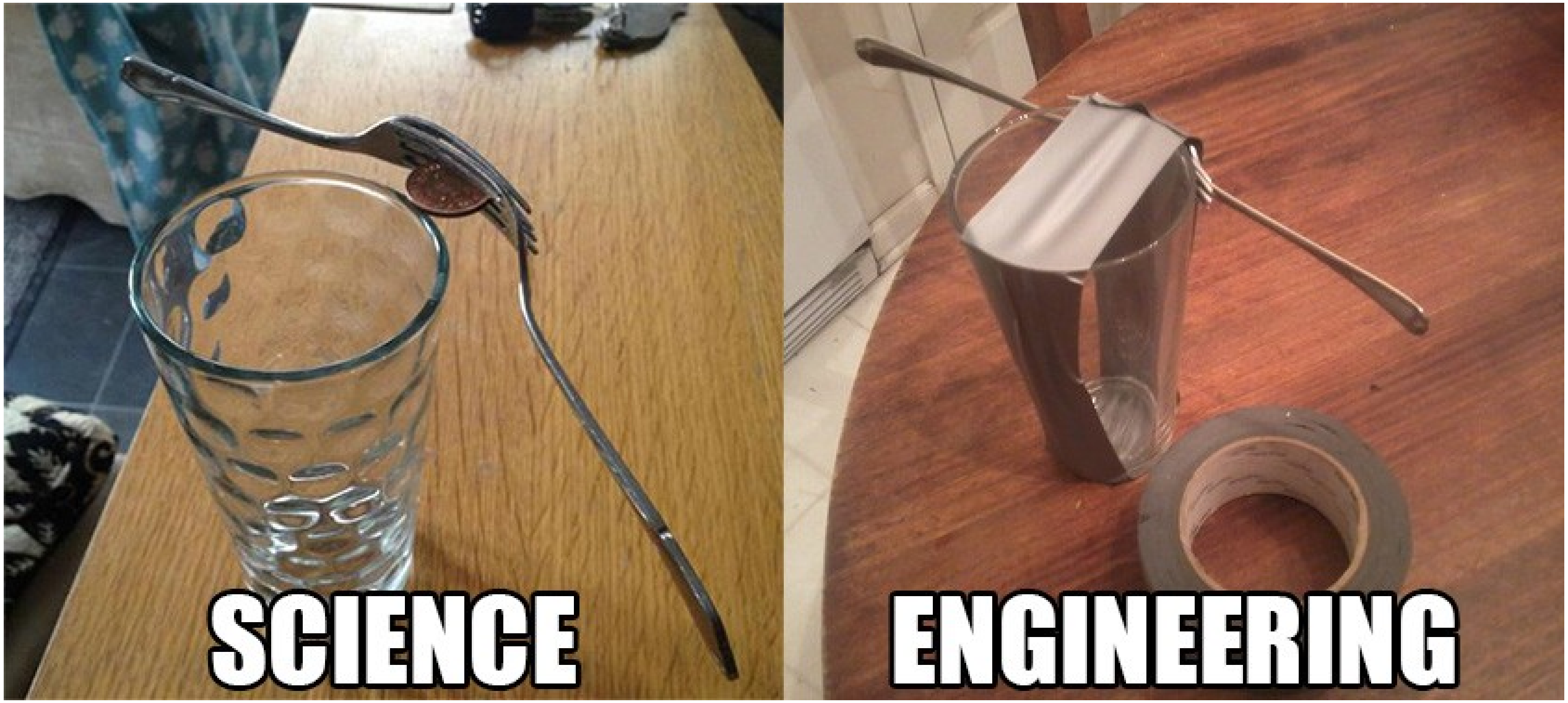
Adaptation

(manual vs automatic)

* for RB systems: add more rules
* for ML systems: online learning algorithms
* for DL systems: transfer learning

Productization

Delivering your NLP application to the users



Product variants

* end-user product
  (web, mobile, desktop)
* internal feature
* API
* library
* scientific paper

Requirements

* speed of
  - processing
  - startup
* memory usage
* storage
* interoperability
* ease of
  - use
  - update

Storage Optimization

WILD Initial model size ~ 1G
(compared to megabytes for CLD,
target: 10-20 MB)

Ways to reduce:

- partly rule-based
- model pruning
- clever algorithms: perfect hash tables, Huffman coding
- model compression, quantization https://arxiv.org/abs/1612.03651

Model Delivery

* pickle & co. 🤮
* JSON, ProtoBuf, …
* HDF5
* Custom gzipped text-based

The Pipeline

“Getting a data pipeline into production for the first time entails dealing with many different moving parts — in these cases, spend more time thinking about the pipeline itself, and how it will enable experimentation, and less about its first algorithm.”
https://medium.com/@neal_lathia/five-lessons-from-building-machine-learning-systems-d703162846ad

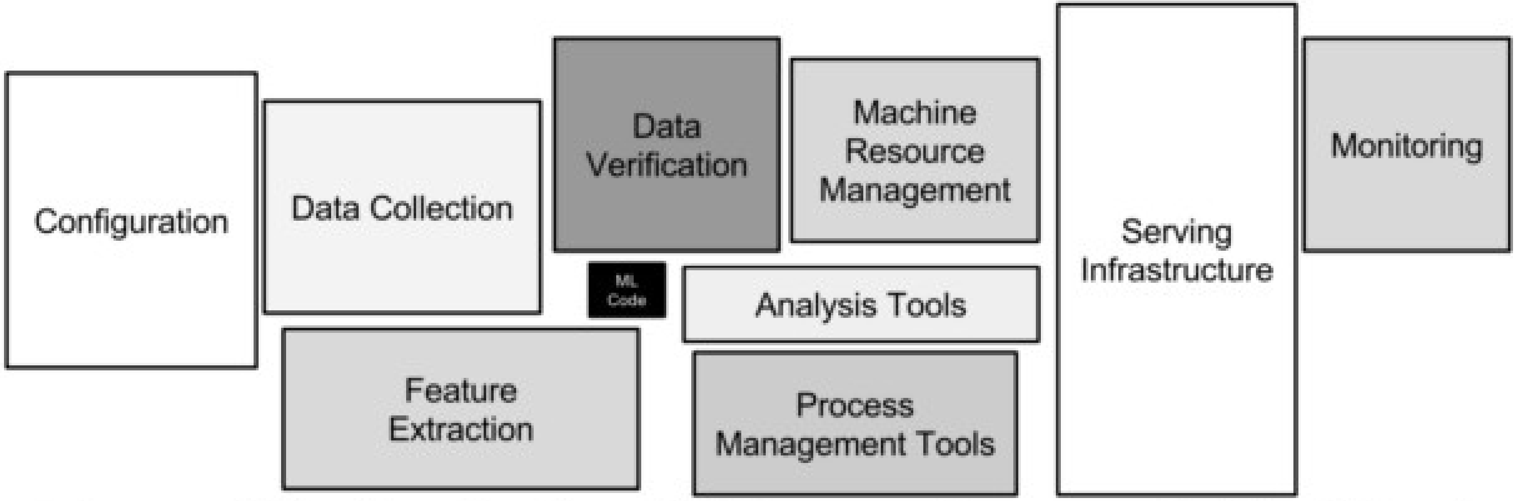


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

Read More

* Top 5 Classification Evaluation Metrics
https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226
* Do people “cheat” by overfitting test data?
https://ehudreiter.com/2020/02/06/cheat-by-overfitting-test-data/
* The Myth of a Strong Baseline:
https://nlpers.blogspot.com/2014/11/the-myth-of-strong-baseline.html
* Native Language Identification
https://www.slideshare.net/frandzi/native-language-identification-brief-review-to-the-state-of-the-art
* Bias in NLP
https://www.slideshare.net/grammarly/grammarly-ainlp-club-1-domain-and-social-bias-in-nlp-case-study-in-language-identification-tim-baldwin-80252288