Social Media & Text Analysis

lecture 3 - natural language processing (part 1): overview and language identification

Follow @cocoweixu

Instructor: Wei Xu

Website: socialmedia-class.org

Natural Language Processing 101

a.k.a.

- Natural Language Processing (NLP)
- Text Analysis
- Computational Linguistics

NLP Publications

- top NLP-specific venues:
  - ACL, NAACL, EACL, EMNLP, COLING (conference)
  - TACL (journal+conference model)
  - CL (journal)

- other venues:
  - NLP field: CoNNL, LREC, RANLP, ACL Workshops …
  - related CS fields: WWW, KDD, AAAI, WSDM, NIPS, ICWSM …
  - related non-CS fields: psychology, linguistics, …

NLP Publications

- ACL Anthology (http://aclweb.org/anthology/)

all NLP conference and journal papers (free!)

ACL Anthology
A Digital Archive of Research Papers in Computational Linguistics

Search the Anthology [ ] via Google | via Searchbench @ DFKI | via AAN @ UMich | via Saffron @ DERI

The ACL Anthology currently hosts over 34,000 papers on the study of computational linguistics and natural language processing. Subscribe to the mailing list to receive announcements and updates to the Anthology.

NEW The beta version of the new ACL Anthology goes live. It will replace this current version of the Anthology as the default version starting 2015 (don't worry we will still maintain both for some duration for handover).

NEW June 2015: The June issue of *Computational Linguistics* journal is now available on the ACL Anthology.

ACL events

CL: Intro FS MT&CL 74-79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 UPDATED 15
TACL: 15 14 13
ACL: Intro 79 80 81 82 83 84* 85 86 87 88 89 90 91 92 93 94 95 96 97* 98* 99 00 01 02 03 04 05 06* 07 08* 09* 10 11 12 13 14
EACL: Intro 83 85 87 89 91 93 95 97* 99 03 06 09 12 14
NAACL: Intro 00* 01 03 04 06* 07* 09* 10* 12* 13* 15*
EMNLP: 96 97 98 99 00 01 02 03 04 05 06 07* 08 09 10 11 12* 13 14
CoNLL: 97 98 99 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14
***Sem/ SemEval:** 98 01 04 07 10 12 13 14 15
ANLP: Intro 83 88 92 94 97 00*
Workshops: 90 91 93 94 95 96 97 98 99 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
SIGs: ANN BIOMED DAT DIAL FSM GEN HAN HUM LEX MEDIA MOL MT NLL PARSE MORPHON SEM SEMITIC SLPAT WAC

Other Events

COLING: 65 67 69 73 80 82 84* 86 88 90 92 94 96 98* 00 02 04 06* 08 10 12 14
HLT: 86 89 90 91 92 93 94 01 03* 04* 05 06* 07* 08* 09* 10* 12* 13* 15*
IJCNLP: 05 08 09* 11 13
LREC: 00 02 04 06 08 10 12 14
PACLIC 95 96 98 99 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14
Rocling Intro 88 89 90 91 92 93 94 95 96 97 98 99 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14
TINLAP: 75 78 87
Donors Needed: COLING-65, any missing COLING

ALTA Intro 03 04 05 06 07 08 09 10 11 12 13 14
RANLP 09 11 13
JEP/TALN/RECITAL 12 13 14
MUC: 91 92 93 95 98
Tipster: 93 96 98
In Progress: Finite String

ACL'14 at A Glance

- The Annual Meeting of the Association for Computational Linguistics
- Duration:
  - tutorials (1 day)
  - main conference (3 days)
  - workshops (2 days)
- Attendance of 1300+ people
- Papers:
  - 1,123 submissions
  - 146 long papers and 129 short papers accepted
  - + 19 TACL papers
  - 159 oral and 145 poster presentations

Popular Areas

- Machine Translation
- Tagging/Chunking/Syntax/Parsing
- Semantics
- Information Extraction / Text Mining
- Sentiment Analysis
- Others: Summarization, Generation, Q&A, Discourse Analysis, Spoken Language, …

Domain/Genre

- NLP is often designed for one domain (in-domain), and may not work well for other domains (out-of-domain).

- Why?

News
Blogs
Wikipedia
Forums
Comments
Twitter
…

Domain/Genre

- How different?

Corpus	Word length	Sentence length
TWITTER-1	3.8±2.4	9.2±6.4
TWITTER-2	3.8±2.4	9.0±6.3
COMMENTS	3.9±3.2	10.5±10.1
FORUMS	3.8±2.3	14.2±12.7
BLOGS	4.1±2.8	18.5±24.8
WIKIPEDIA	4.5±2.8	21.9±16.2
BNC	4.3±2.8	19.8±14.5

Source: Baldwin et al.
"How Noisy Social Media Text, How Diffrnt Social Media Sources?" IJCNLP 2013

Domain/Genre

- How different?

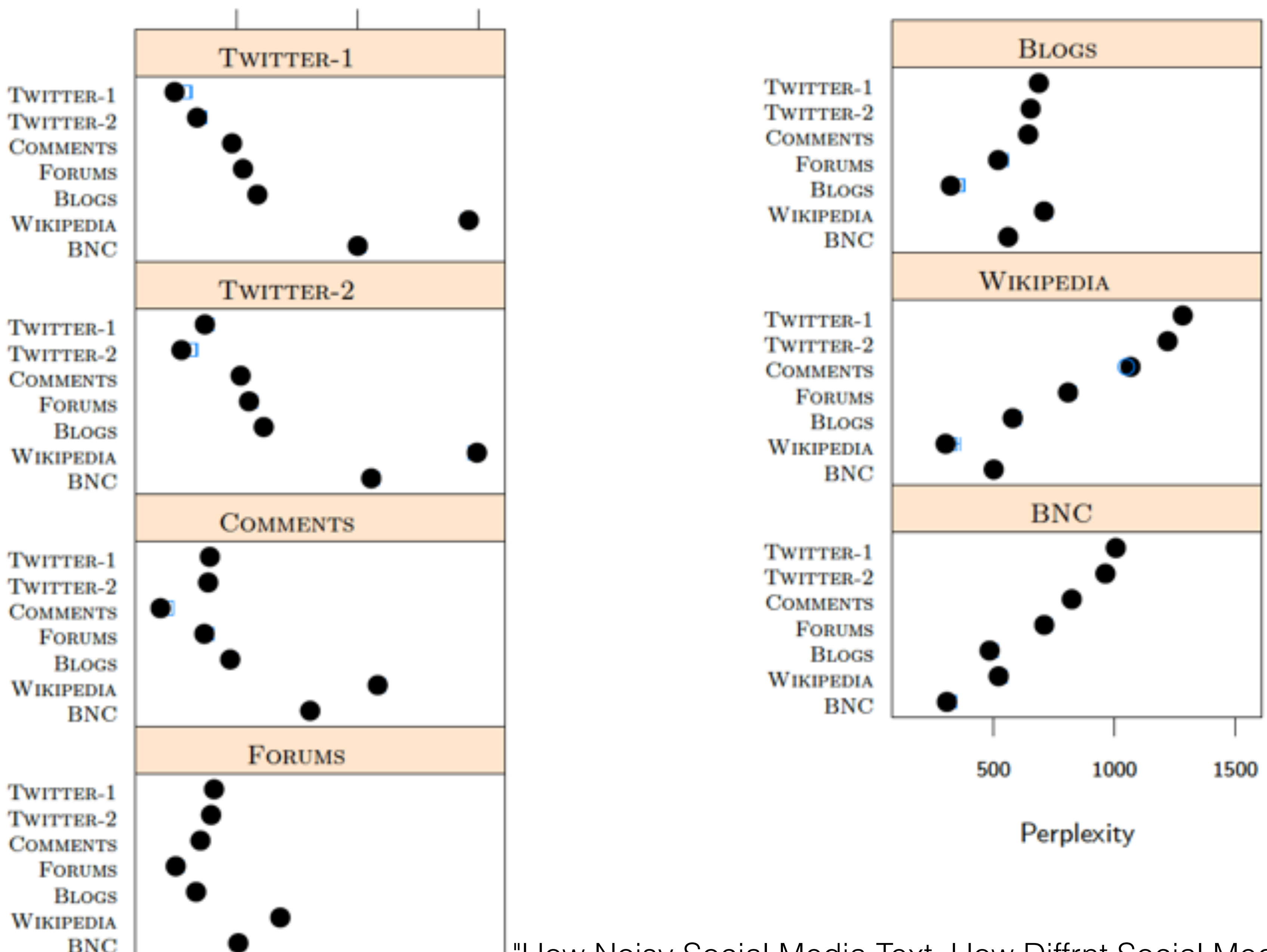
out-of-vocabulary

Corpus	Word length	Sentence length	%OOV
TWITTER-1	3.8±2.4	9.2±6.4	**24.6**
TWITTER-2	3.8±2.4	9.0±6.3	**24.0**
COMMENTS	3.9±3.2	10.5±10.1	**19.8**
FORUMS	3.8±2.3	14.2±12.7	**18.1**
BLOGS	4.1±2.8	18.5±24.8	**20.6**
WIKIPEDIA	4.5±2.8	21.9±16.2	**19.0**
BNC	4.3±2.8	19.8±14.5	**16.9**

Source: Baldwin et al.
"How Noisy Social Media Text, How Diffrnt Social Media Sources?" IJCNLP 2013

Domain/Genre

- How similar?
  Twitter ≡ Comments < Forums < Blogs < BNC < Wikipedia



Source: Baldwin et al.
"How Noisy Social Media Text, How Diffrnt Social Media Sources?" IJCNLP 2013

Domain/Genre

- What to do?
  - robust tools/models that works across domains
  - specific tools/models for Twitter data only — many techniques/algorithms are useful elsewhere

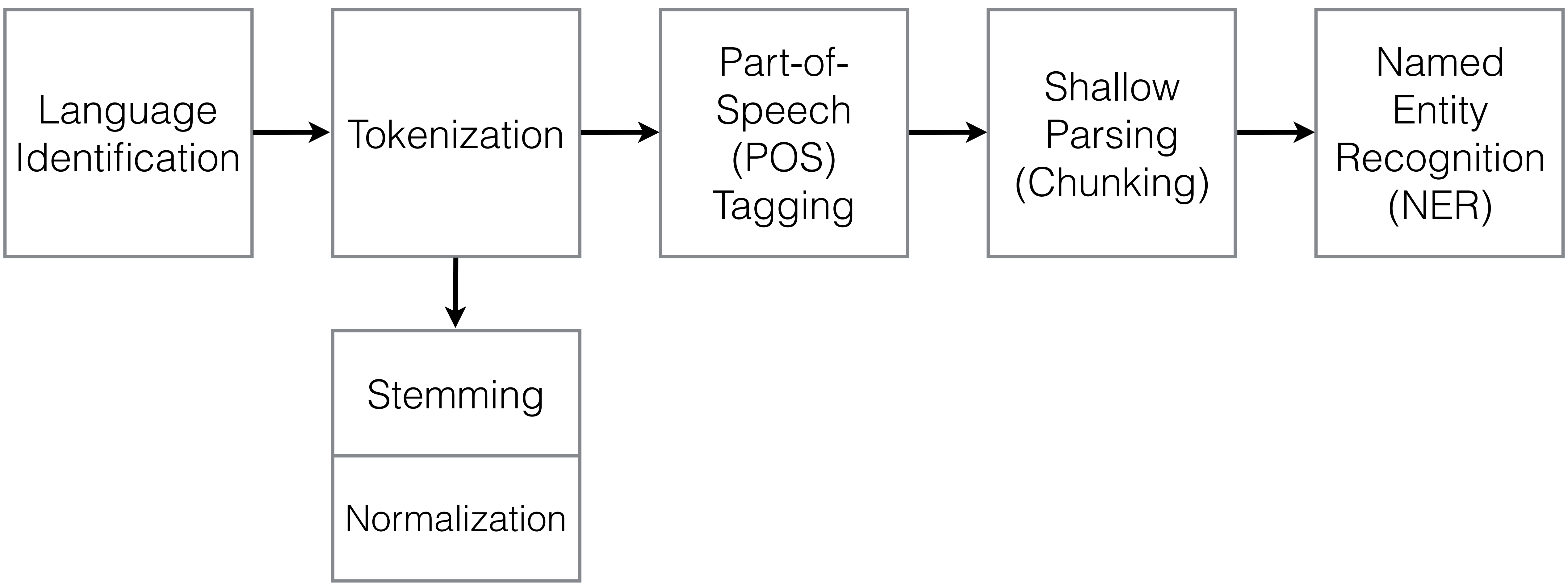
(we will see examples of both in the class)

Domain/Genre

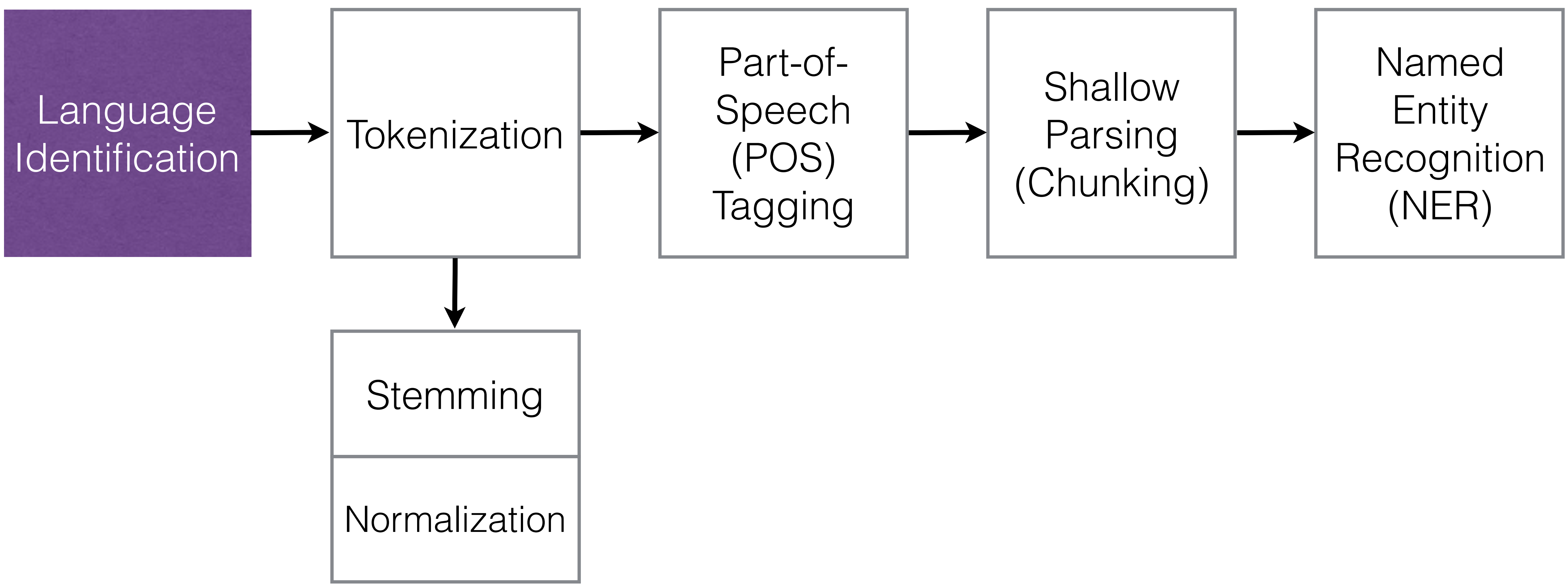
- Why so much Twitter?
  - publicly available (vs. SMS, emails)
  - large amount of data
  - large demand for research/commercial purpose
  - too different from well-edited text (which most NLP tools have been made for)

NLP Pipeline

NLP Pipeline

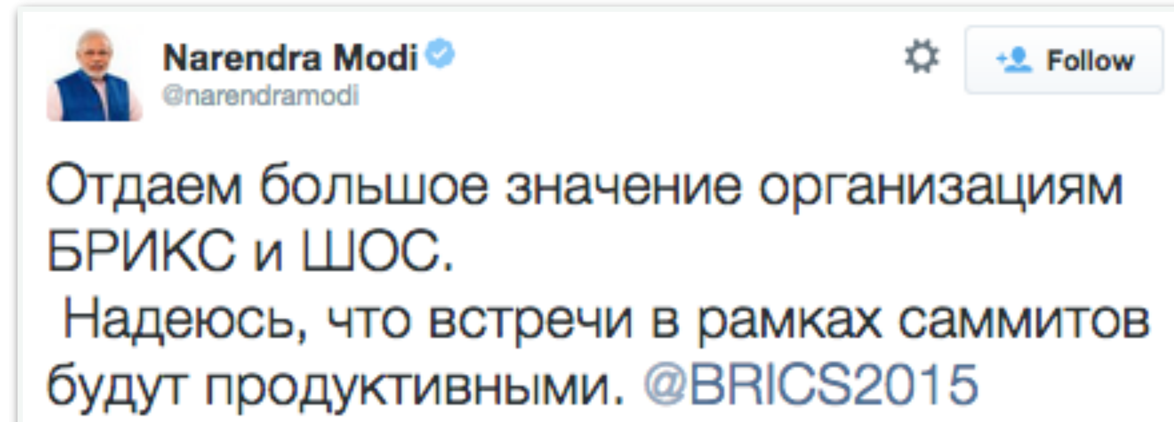


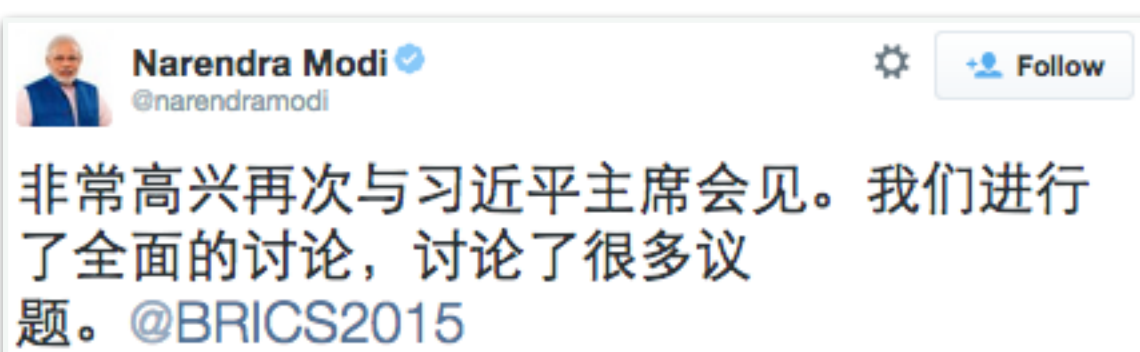
NLP Pipeline

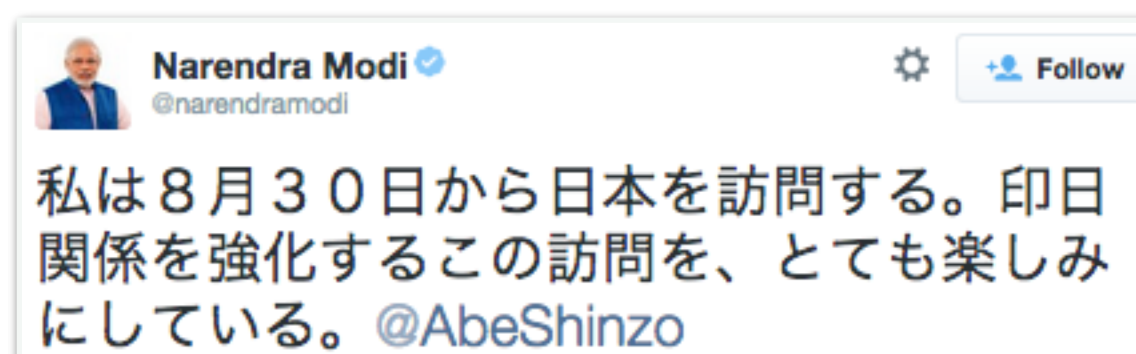


Language Identification

(a.k.a Language Detection)





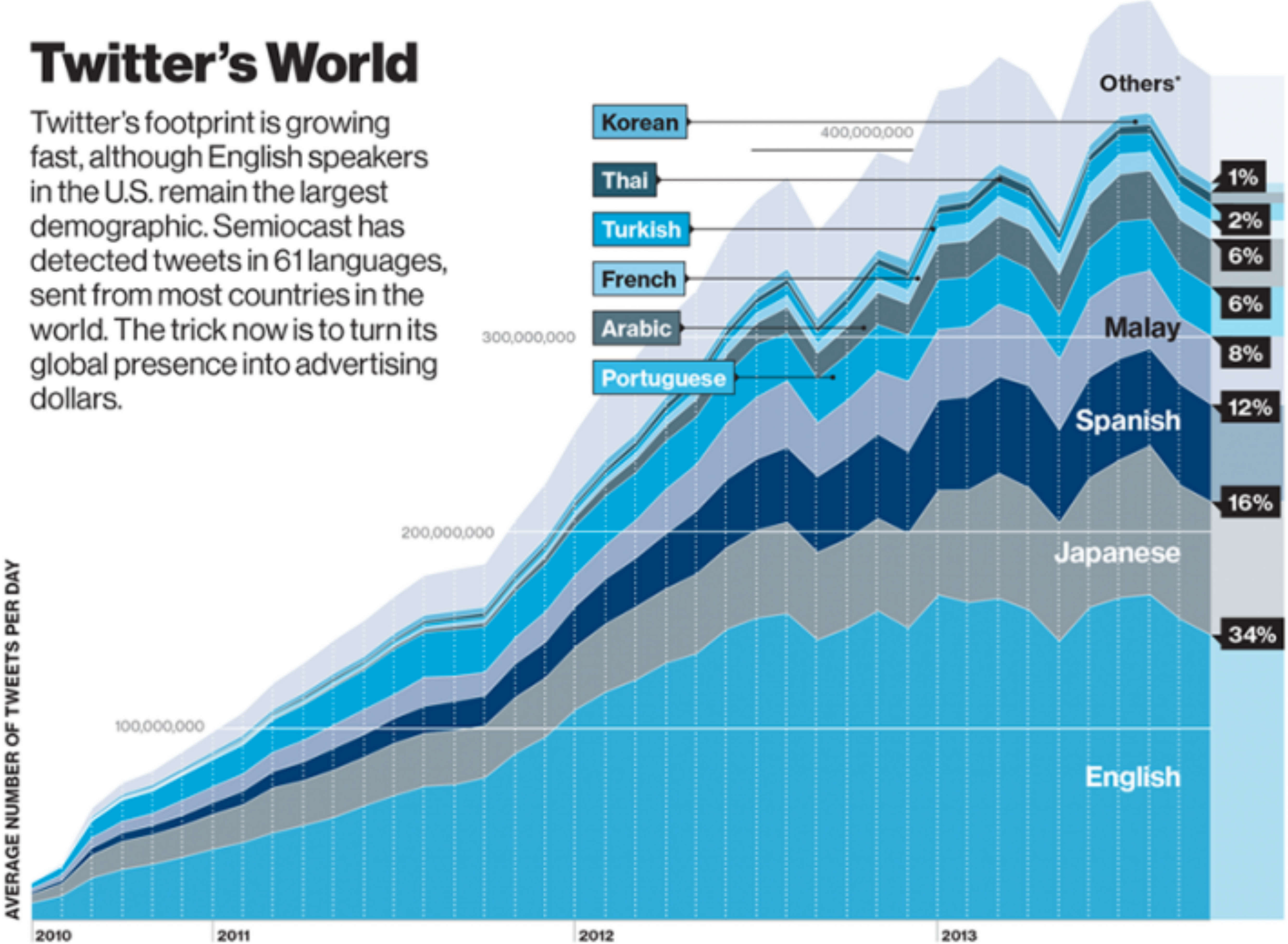


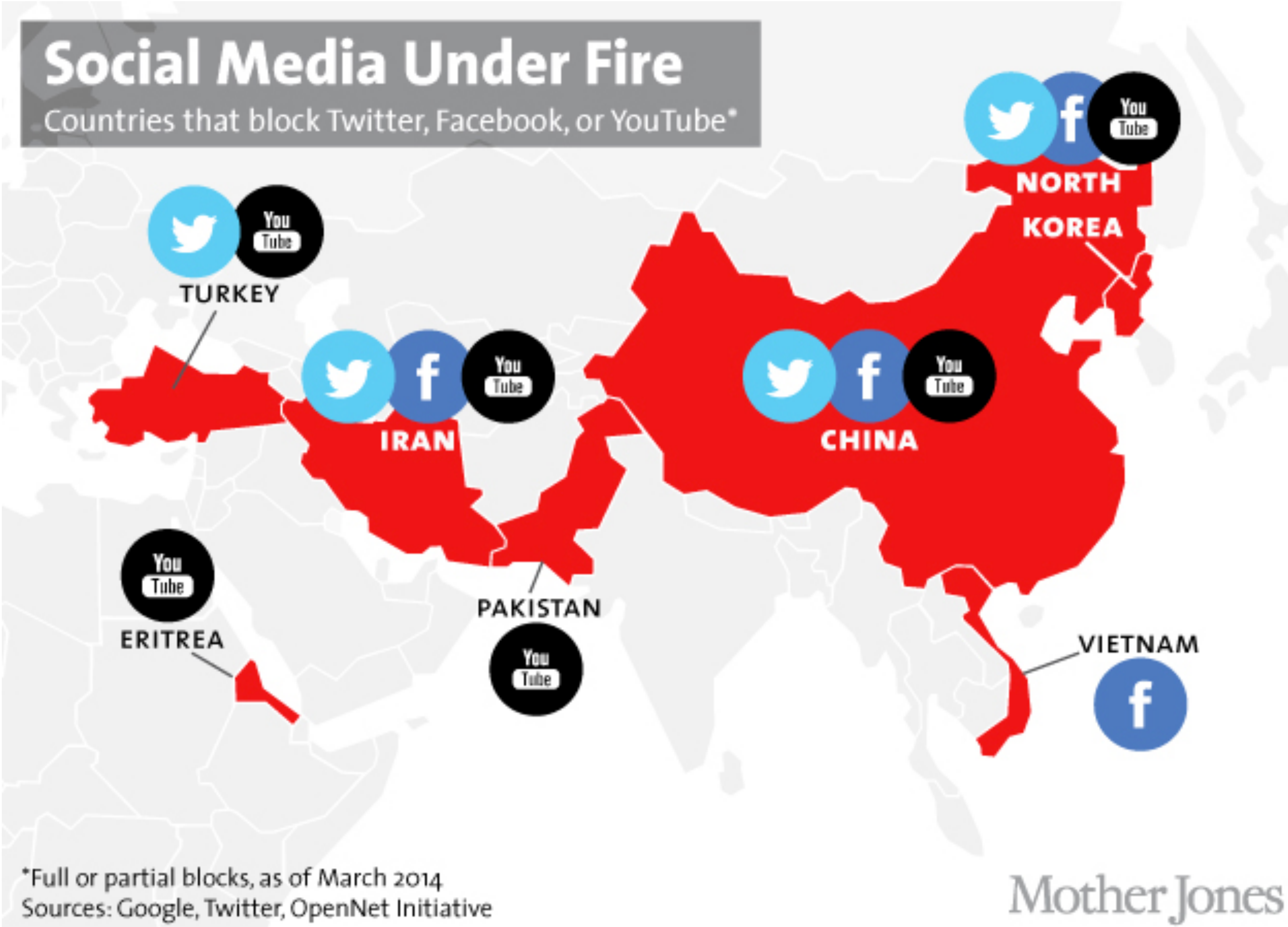
LangID: why needed?

- Twitter is highly multilingual
- But NLP is often monolingual

Twitter's World

Twitter's footprint is growing fast, although English speakers in the U.S. remain the largest demographic. Semiocast has detected tweets in 61 languages, sent from most countries in the world. The trick now is to turn its global presence into advertising dollars.

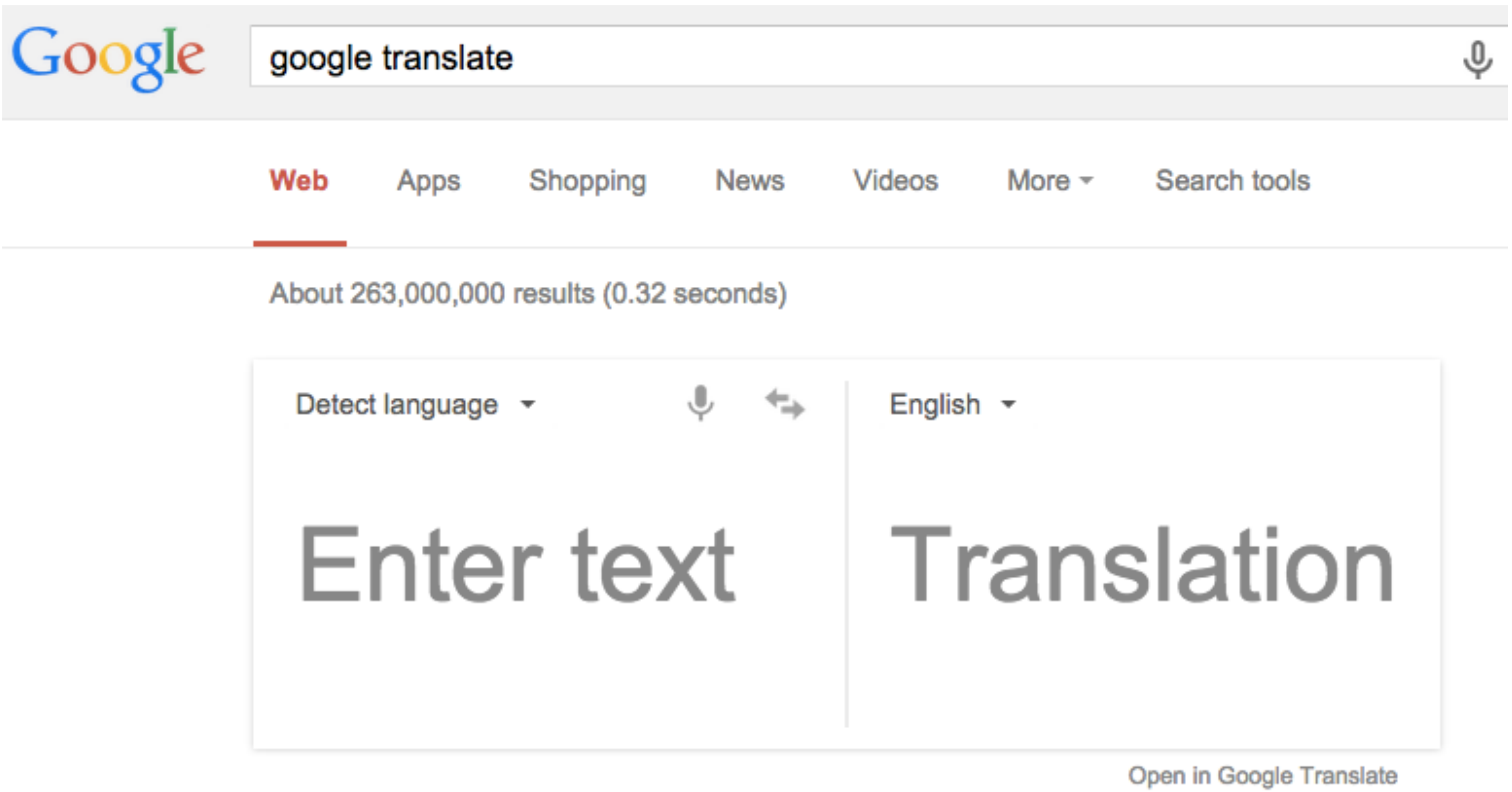




新浪微博
weibo.com

known as the “Chinese Twitter”
120 Million Posts / Day

LangID: Google Translate



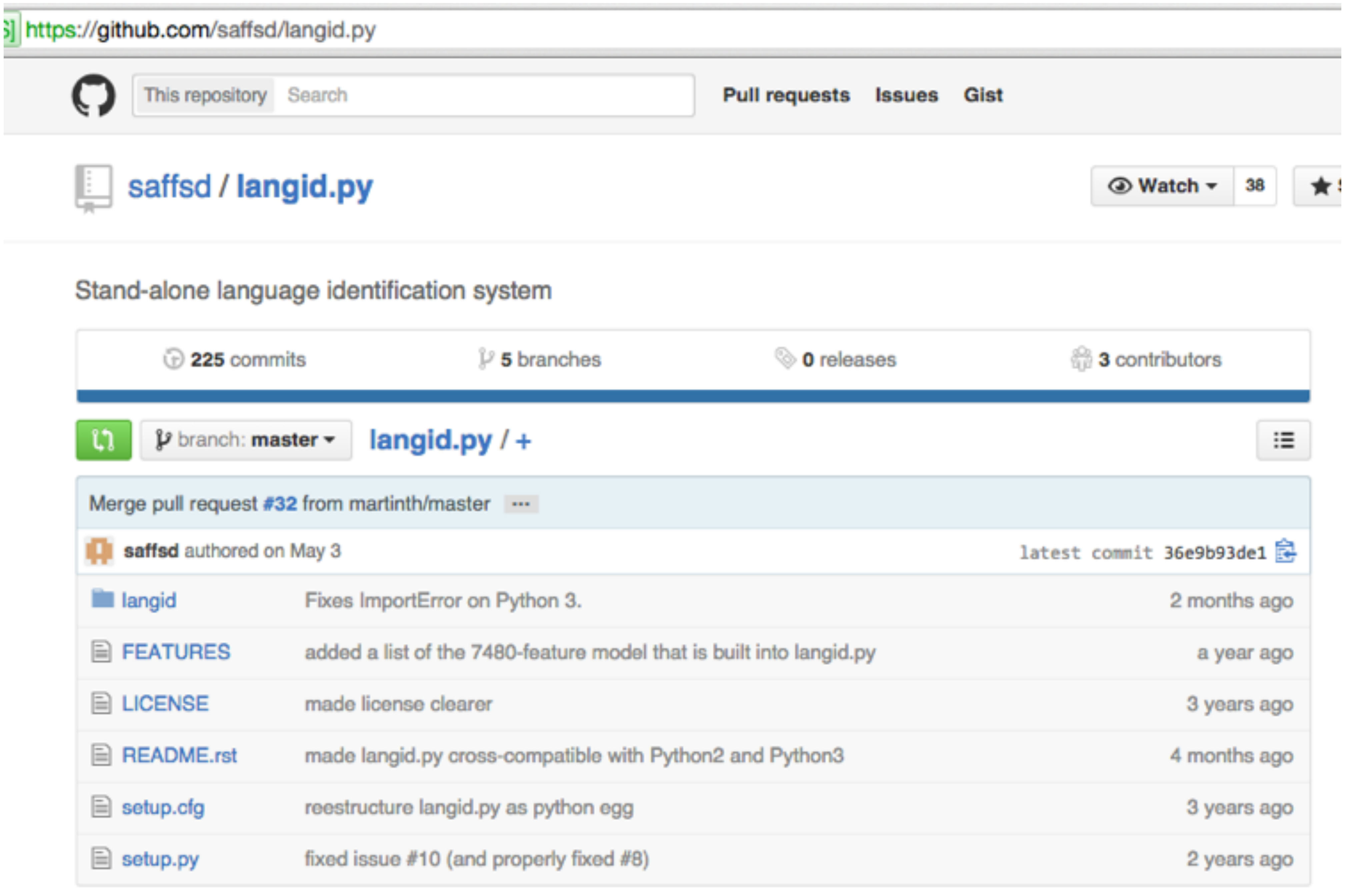
LangID: Twitter API

- introduced in March 2013
- uses two-letter ISO 639-1 code


```
"status": {
    "created_at": "Tue Oct 30 21:12:37 +0000 2012",
    "id": 263387958047027200,
    "id_str": "263387958047027200",
    "text": "Better late than never, statuses/retweets_of_me is joining the API v1.1
method roster: https://t.co/jYz3MJnb ^TS",
    "geo": null,
    "coordinates": null,
    "place": null,
    "filter_level": "medium",
    "lang": "en",
    ...
}
```


language detection

LangID Tool: langid.py



LangID Tool: langid.py

```
python
Python 2.7.2+ (default, Oct  4 2011, 20:06:09)
[GCC 4.6.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import langid
>>> langid.classify("I do not speak english")
('en', 0.57133487679900674)
>>> langid.set_languages(['de','fr','it'])
>>> langid.classify("I do not speak english")
('it', 0.99999835791478453)
>>> langid.set_languages(['en','it'])
>>> langid.classify("I do not speak english")
('en', 0.99176190378750373)
```

LangID:

A Classification Problem

- Input:
  - a document $\boldsymbol{d}$
  - a fixed set of classes $\boldsymbol{C = \{c_1, c_2, \ldots, c_j\}}$

- Output:
  - a predicted class $\boldsymbol{c} \in \boldsymbol{C}$

Classification Method:
Hand-crafted Rules

- Keyword-based approaches do not work well for language identification:
  - poor recall
  - cognate words
  - expensive to build large dictionaries for all different languages

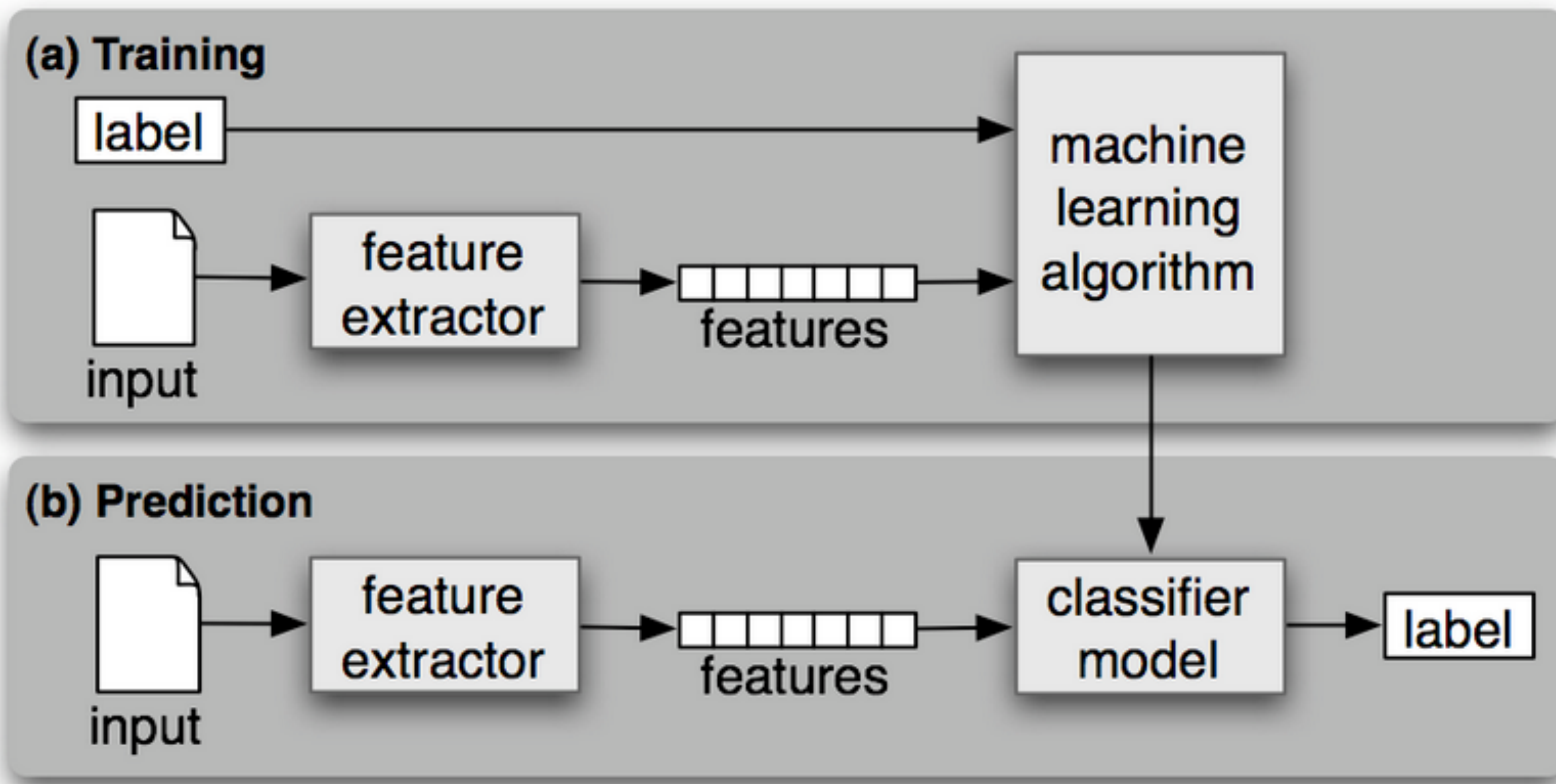
Classification Method:

Supervised Machine Learning

- Input:
  - a document $\boldsymbol{d}$
  - a fixed set of classes $\boldsymbol{C = \{c_1, c_2, \ldots, c_j\}}$
  - a training set of $\boldsymbol{m}$ hand-labeled documents $\boldsymbol{(d_1, c_1), \ldots, (d_m, c_m)}$
- Output:
  - a learned classifier $\boldsymbol{\gamma : d \rightarrow c}$

Classification Method:

Supervised Machine Learning



Source: NLTK Book

Classification Method:
Supervised Machine Learning

- Naïve Bayes
- Logistic Regression
- Support Vector Machines (SVM)
- …

Classification Method:

Supervised Machine Learning

- **Naïve Bayes**
- Logistic Regression
- Support Vector Machines (SVM)
- …

Naïve Bayes

- For a document ***d***, find the most probable class ***c***:

$$c_{MAP} = \underset{c \in C}{\arg\max}\, P(c \mid d)$$

maximum a posteriori

Source: adapted from Dan jurafsky

Naïve Bayes

- For a document ***d***, find the most probable class ***c***:

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(c \mid d)$$

$$= \underset{c \in C}{\operatorname{argmax}} \frac{P(d \mid c) P(c)}{P(d)}$$

← **Bayes Rule**

Source: adapted from Dan jurafsky

Naïve Bayes

- For a document ***d***, find the most probable class ***c***:

$$c_{MAP} = \underset{c \in C}{\arg\max}\, P(c \mid d)$$

$$= \underset{c \in C}{\arg\max}\, \frac{P(d \mid c) P(c)}{P(d)}$$ ← **Bayes Rule**

$$= \underset{c \in C}{\arg\max}\, P(d \mid c) P(c)$$ ← **drop the denominator**

Source: adapted from Dan jurafsky

Naïve Bayes

- document ***d*** represented as features $\boldsymbol{t_1, t_2, \ldots, t_n}$:

$$c_{MAP} = \underset{c \in C}{\arg\max}\, P(d \mid c) P(c)$$

$$= \underset{c \in C}{\arg\max}\, P(t_1, t_2, \ldots, t_n \mid c) P(c)$$

Source: adapted from Dan jurafsky

Naïve Bayes

- document ***d*** represented as features ***t_1, t_2, …, t_n***:

$$c_{MAP} = \underset{c \in C}{\arg\max}\, P(t_1, t_2, \ldots, t_n \mid c)\underbrace{P(c)}_{\text{prior}}$$

how often does this class occur? — simpe count

Source: adapted from Dan jurafsky

Naïve Bayes

- document ***d*** represented as features ***t₁, t₂, …, tₙ***:

$$c_{MAP} = \underset{c \in C}{\arg\max}\, \underbrace{P(t_1, t_2, \ldots, t_n \mid c)}_{\text{likelihood}}\, \underbrace{P(c)}_{\text{prior}}$$

$O(|T|^n \cdot |C|)$ parameters

n = number of unique n-gram tokens

— need to make simplifying assumption

Source: adapted from Dan jurafsky

Naïve Bayes

- **Conditional Independence Assumption**:

  features ***P(t_i|c)*** are independent given the class ***c***

$$P(t_1, t_2, \ldots, t_n \mid c)$$
$$= P(t_1 \mid c) \cdot P(t_2 \mid c) \cdot \ldots \cdot P(t_n \mid c)$$

Source: adapted from Dan jurafsky

Naïve Bayes

- For a document ***d***, find the most probable class ***c***:

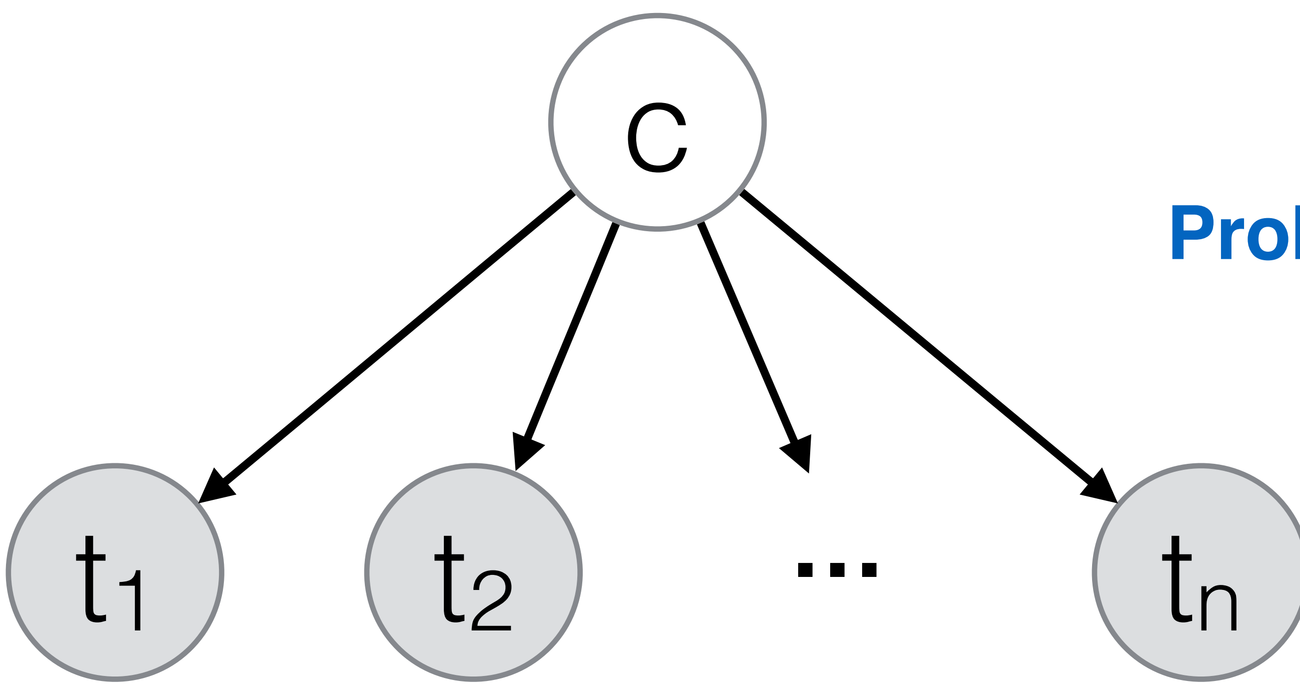
$$c_{MAP} = \underset{c \in C}{\arg\max}\, P(t_1, t_2, \ldots, t_n \mid c) P(c)$$

$$c_{NB} = \underset{c \in C}{\arg\max}\, P(c) \prod_{t_i \in d} P(t_i \mid c)$$

Source: adapted from Dan jurafsky

Naïve Bayes

$$c_{NB} = \underset{c \in C}{\arg\max}\, P(c) \prod_{t_i \in d} P(t_i \mid c)$$



Probabilistic Graphical Model

Variations of Naïve Bayes

$$c_{MAP} = \underset{c \in C}{\arg\max}\, \boxed{P(d \mid c)}\, P(c)$$

- different assumptions on distributions of feature:
  - *Multinomial: discrete features*
  - *Bernoulli: discrete feature (binary)*
  - *Gaussian: continuous features*

Source: adapted from Dan jurafsky

Variations of Naïve Bayes

$$c_{MAP} = \underset{c \in C}{\arg\max}\, \boxed{P(d \mid c)}\, P(c)$$

- different assumptions on distributions of feature:
  - ***Multinomial***: *discrete features*
  - *Bernoulli: discrete feature (binary)*
  - *Gaussian: continuous features*

Source: adapted from Dan jurafsky

LangID features

- n-grams features:
  - 1-grams:
    "the" "following" "Wikipedia"
    "en" "español" …
  - 2-grams:
    "the following" "following is"
    "Wikipedia en" "en español" …
  - 3-grams:
    ….

English

The following is a list of words that occur in both Modern English and Modern Spanish, but which are pronounced differently and may have different meanings in each language.
…

Spanish

Wikipedia en español es la edición en idioma español de Wikipedia. Actualmente cuenta con 1 185 590 páginas válidas de contenido y ocupa el décimo puesto en esta estadística entre
…

Bag-of-Words Model

- **positional independence assumption**:
  - features are the words occurring in the document and their value is the number of occurrences
  - word probabilities are position independent

Naïve Bayes

$$c_{NB} = \underset{c \in C}{\operatorname{arg\,max}}\, P(c) \prod_{t_i \in d} P(t_i \mid c)$$

- Learning the Multinomial Naïve Bayes model simply uses the frequencies in the training data:

$$\hat{P}(c) = \frac{count(c)}{\sum_{c_j \in C} count(c_j)} \qquad \hat{P}(t \mid c) = \frac{count(t, c)}{\sum_{t_i \in V} count(t_i, c)}$$

Source: adapted from Dan jurafsky

Naïve Bayes

	Doc	Words	Class
Training	1	English Wikipedia editor	en
	2	free English Wikipedia	en
	3	Wikipedia editor	en
	4	español de Wikipedia	es
Test	5	Wikipedia español el	?

$$\hat{P}(c) = \frac{count(c)}{\sum_{c_j \in C} count(c_j)}$$

P(en)=3/4   P(sp)=1/4

$$\hat{P}(t \mid c) = \frac{count(t,c)}{\sum_{t_i \in V} count(t_i,c)}$$

P("Wikipedia" | en) = 3/8 , P("Wikipedia" | es) = 1/3
P("español" | en) = 0/8 , P("español" | es) = 1/3
P("el" | en) = 0/8 , P("el" | es) = 0/3

P(en|doc5) = 3/4×3/8×0/8×0/8 = 0
P(es|doc5) = 1/4×2/9×1/3×0/3 = 0

Naïve Bayes

- What if the word "el" doesn't occur in the training documents that labeled as Spanish(es)?

$$\hat{P}("el" \mid es) = \frac{count("el", es)}{\sum_{t \in V} count(t, es)} = 0$$

- To deal with 0 counts, use add-one or Laplace smoothing:

$$\hat{P}(t \mid c) = \frac{count(t, c)}{\sum_{t_i \in V} count(t_i, c)} \longrightarrow \hat{P}(t \mid c) = \frac{count(t, c) + 1}{\sum_{t_i \in V} count(t_i, c) + |V|}$$

Source: adapted from Dan jurafsky

Naïve Bayes

	Doc	Words	Class
Training	1	English Wikipedia editor	en
	2	free English Wikipedia	en
	3	Wikipedia editor	en
	4	español de Wikipedia	sp
Test	5	Wikipedia español el	?

$$\hat{P}(c) = \frac{count(c)}{\sum_{c_j \in C} count(c_j)}$$

P(en)=3/4 P(sp)=1/4

$$\hat{P}(t \mid c) = \frac{count(t,c)}{\sum_{t_i \in V} count(t_i,c)}$$

P("Wikipedia" |en) = 3+1/8+6 , P("Wikipedia" |sp) = 1+1/3+6
P("español" |en) = 0+1/8+6 , P("español" |sp) = 1+1/3+6
P("el" |en) = 0+1/8+6 , P("el" |sp) = 0+1/3+6

P(en|doc5) = 3/4×4/14×1/14×1/14 = 0.00109
P(sp|doc5) = 1/4×2/9×2/9×1/9 = 0.00137

Naïve Bayes

- Pros:
    - simple (no iterative learning)
    - fast and light-weighted
    - less parameter, so need less training data
    - even if the NB assumption doesn't hold, a NB classifier still often performs surprisingly well in practice (e.g. text classification)

- Cons
    - assumes independence of features
    - can't model dependencies/structures (e.g. correlated features)

LangID Tool: langid.py

```
python
Python 2.7.2+ (default, Oct  4 2011, 20:06:09)
[GCC 4.6.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import langid
>>> langid.classify("I do not speak english")
('en', 0.57133487679900674)
>>> langid.set_languages(['de','fr','it'])
>>> langid.classify("I do not speak english")
('it', 0.99999835791478453)
>>> langid.set_languages(['en','it'])
>>> langid.classify("I do not speak english")
('en', 0.99176190378750373)
```

LangID Tool: langid.py

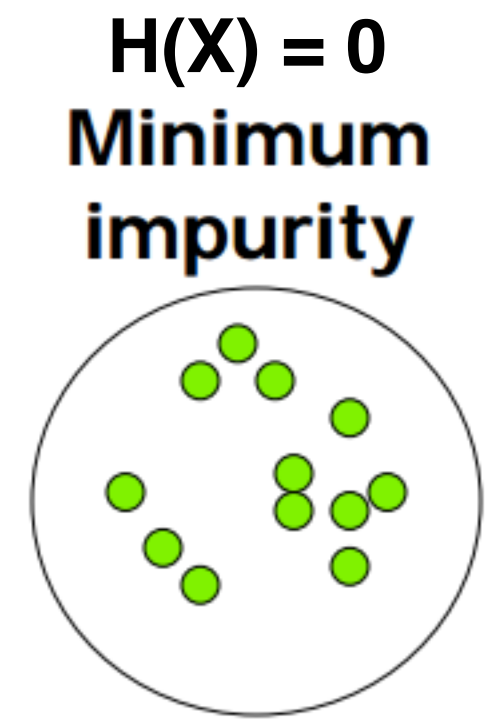
- main techniques:
  - **Multinominal Naïve Bayes**
  - diverse training data from multiple domains (Wikipedia, Reuters, Debian, etc.)
  - plus **feature selection** using **Information Gain (IG)** to choose features that are informative about language, but not informative about domain

Source: Lui and Baldwin "langid.py: An Off-the-shelf Language Identification Tool" ACL 2012

Information Gain

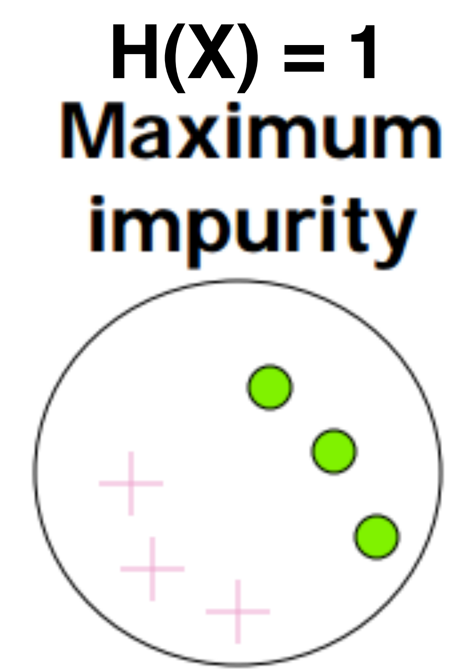
- Information Gain:

$$IG(Y \mid X) = H(Y) - H(Y \mid X)$$

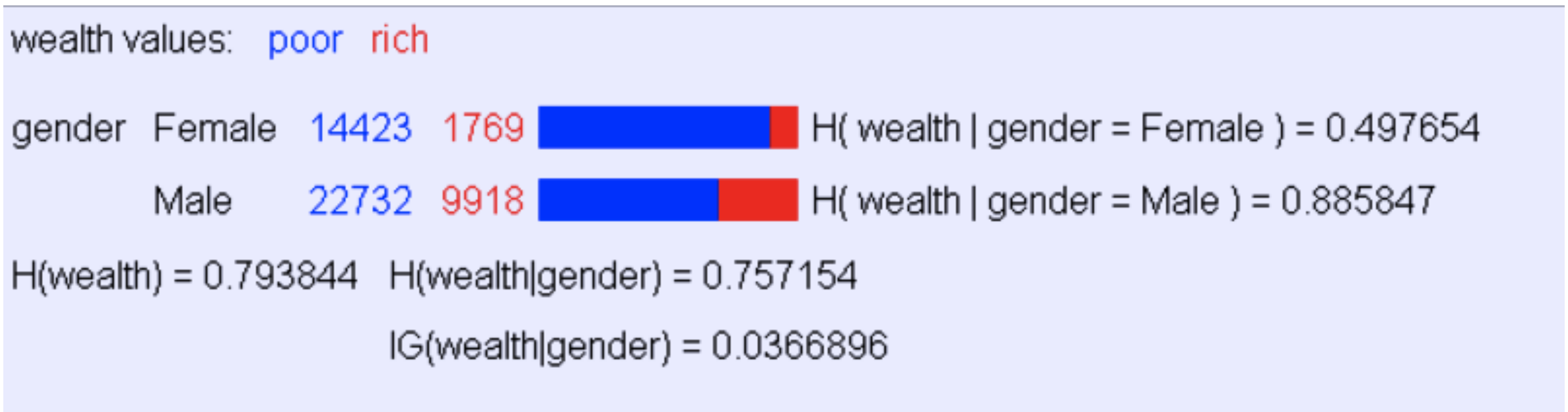


- Entropy:

$$H(X) = -\sum_i P(x_i) \log P(x_i)$$



Information Gain

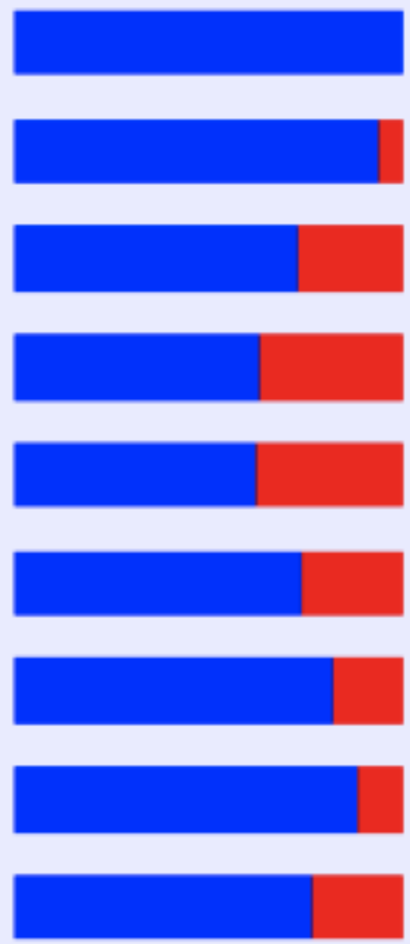


Source: Andrew Moore

Information Gain

wealth values: poor rich

agegroup	10s	2507	3	H(wealth | agegroup = 10s) = 0.0133271
	20s	11262	743	H(wealth | agegroup = 20s) = 0.334906
	30s	9468	3461	H(wealth | agegroup = 30s) = 0.838134
	40s	6738	3986	H(wealth | agegroup = 40s) = 0.951961
	50s	4110	2509	H(wealth | agegroup = 50s) = 0.957376
	60s	2245	809	H(wealth | agegroup = 60s) = 0.834049
	70s	668	147	H(wealth | agegroup = 70s) = 0.680882
	80s	115	16	H(wealth | agegroup = 80s) = 0.535474
	90s	42	13	H(wealth | agegroup = 90s) = 0.788941



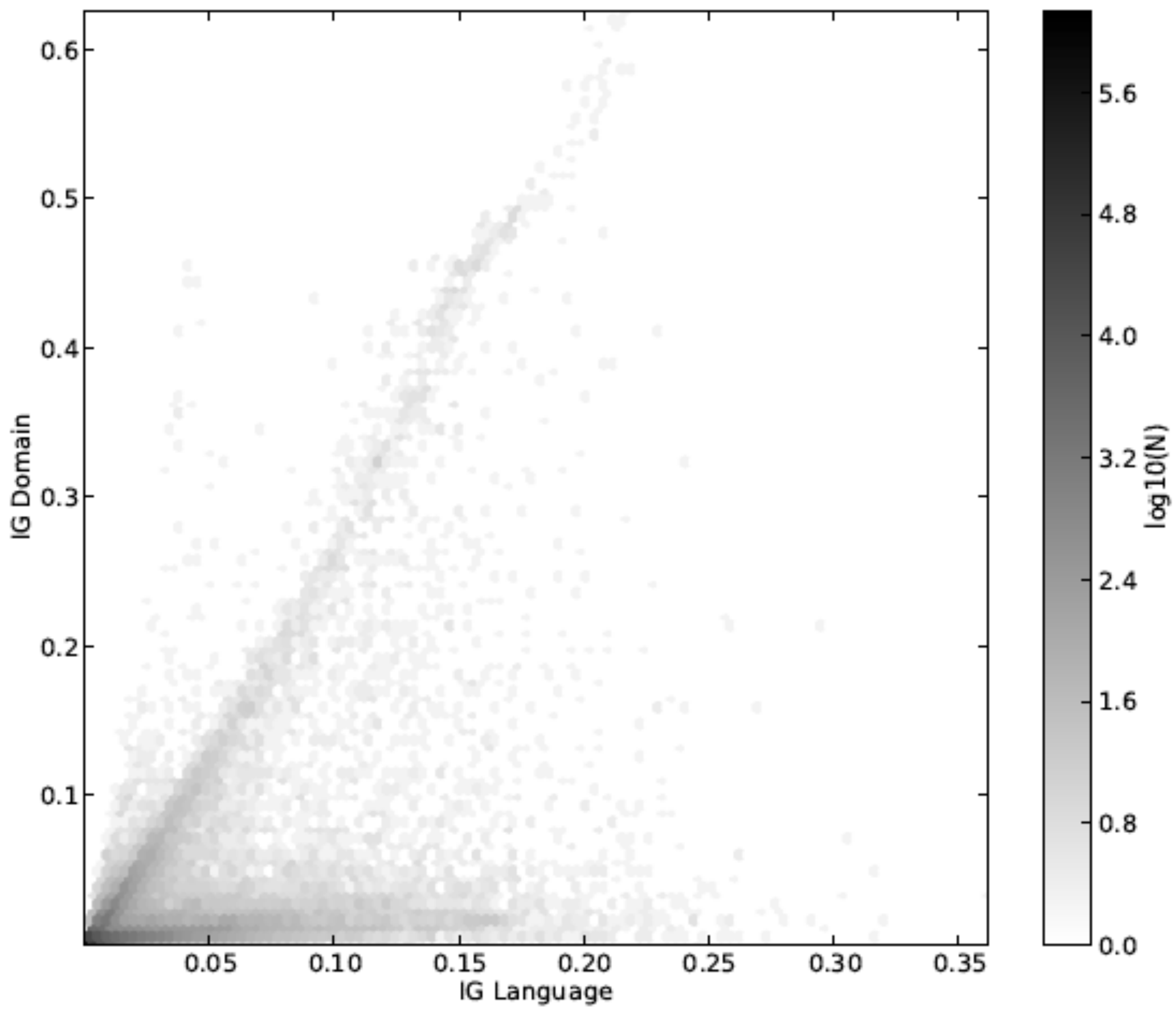
H(wealth) = 0.793844 H(wealth|agegroup) = 0.709463

IG(wealth|agegroup) = 0.0843813

Source: Andrew Moore

LangID Tool: langid.py

- feature selection using Information Gain (IG)



Source: Lui and Baldwin "langid.py: An Off-the-shelf Language Identification Tool" ACL 2012

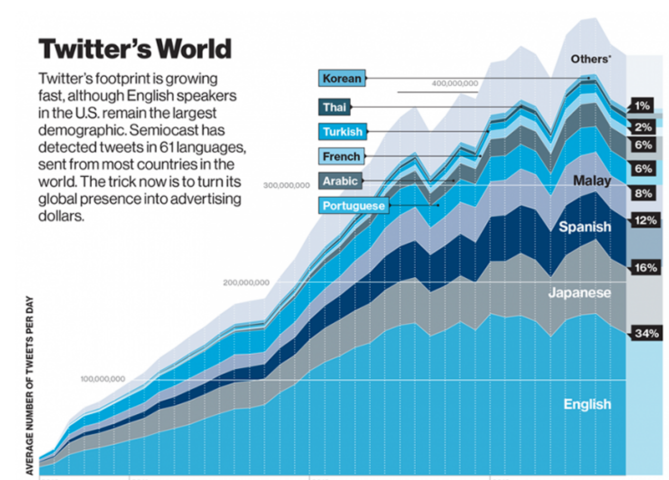
LangID Tool: langid.py

- main advantages:
  - cross-domain (works on all kinds of texts)
  - works for Twitter (accuracy = 0.89)
  - fast (300 tweets/second — 24G RAM)
  - currently supports 97 language
  - retrainable

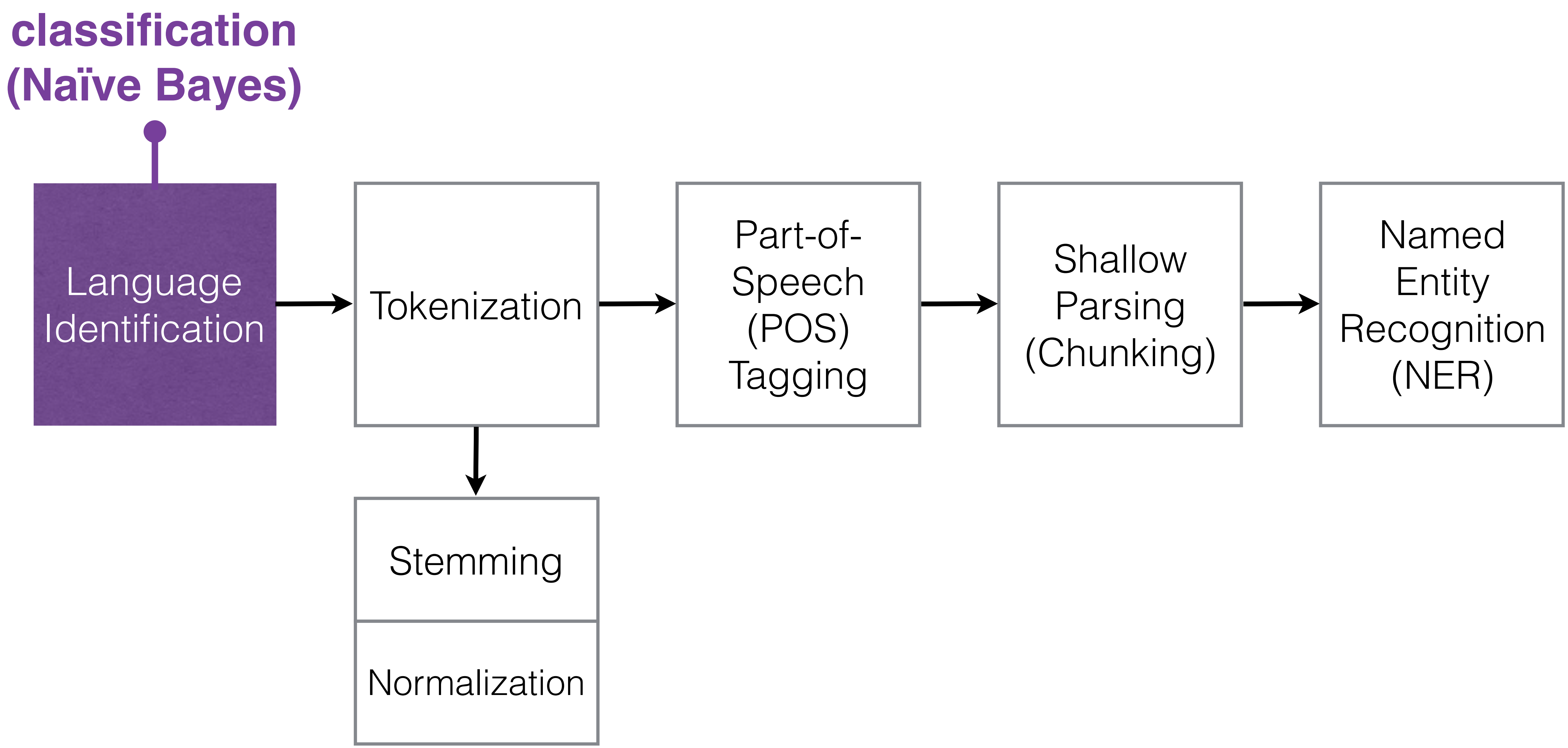
Source: Lui and Baldwin "langid.py: An Off-the-shelf Language Identification Tool" ACL 2012

Homework #2

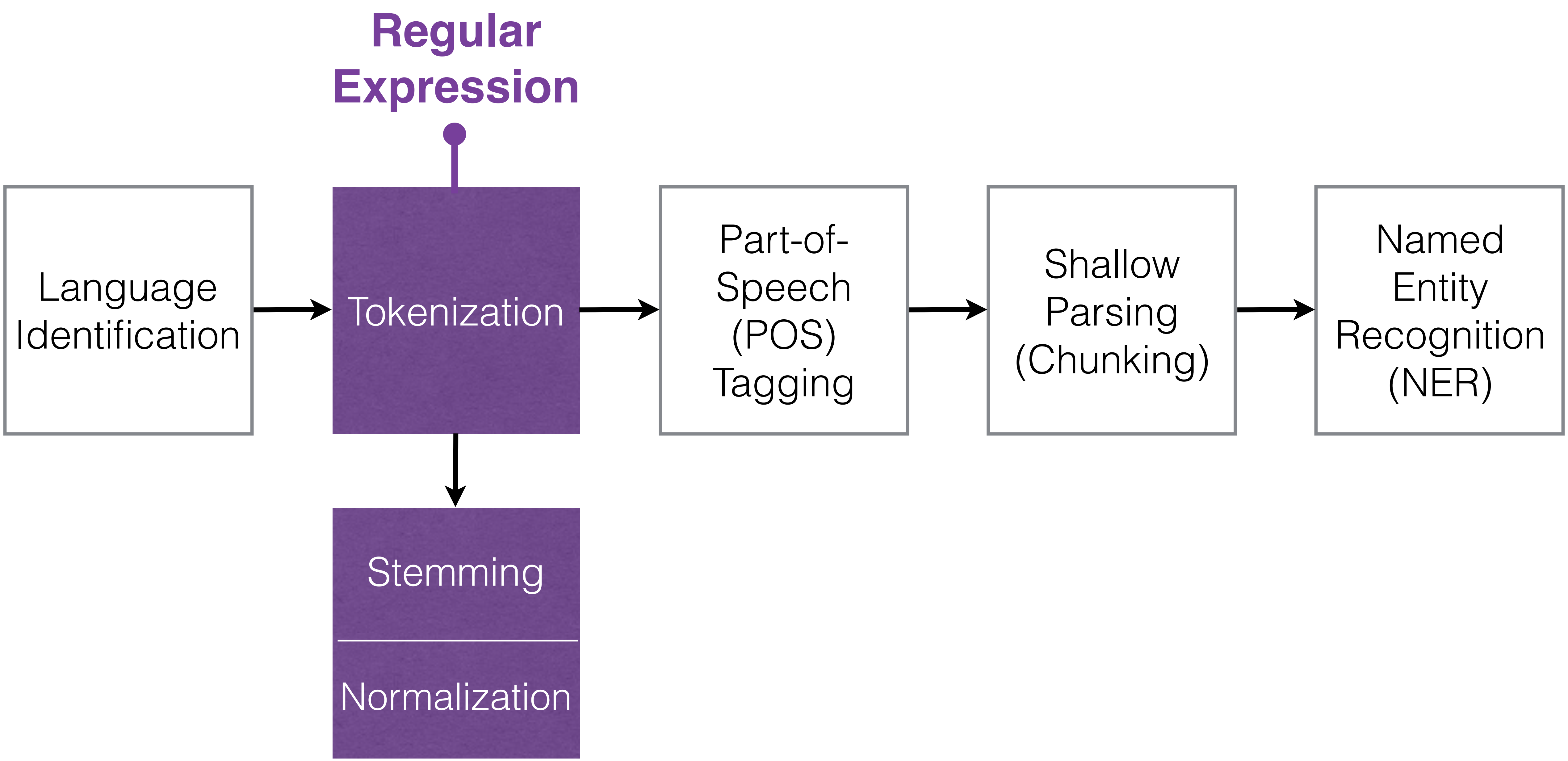
- Get >=10k tweets from Twitter Streaming API
- and check:
  - are all tweets LangID tagged (what %)?
  - how many different language tags?
- then run langid.py and check:
  - how many different language tagged?
  - what % langid.py and Twitter's API agree/disagree?
  - what kind of tweets/languages do they disagree?
- what about tweets in US?
- draw some fancy plots (e.g. language by #tweets)



Summary


classification
(Naïve Bayes)
Language Identification
Tokenization
Part-of-Speech (POS) Tagging
Shallow Parsing (Chunking)
Named Entity Recognition (NER)
Stemming
Normalization

Next Lecture



Thank You!

Follow @cocoweixu

Instructor: Wei Xu

www.cis.upenn.edu/~xwe/

Course Website: socialmedia-class.org