

Social Media & Text Analysis

lecture 3 - Language Identification
(supervised learning and Naive Bayes algorithm)



CSE 5539-0010 Ohio State University
Instructor: Wei Xu
Website: socialmedia-class.org

In-class Presentation

- a 8-minute presentation plus 2-minute Q&A (20 points)
 - A Social Media Platform or a NLP Research Group
 - Pairing up (2 students collaboration)
- Sign up now!

Social Media & Text Analytics

Syllabus

Twitter API Tutorial

Homework ▾



Social Media Map for
2016

Survey a Social Media Platform or a NLP Research Group : In-class Presentation (20 points)

You will pair together (2 students) and give a 8-minute presentation (plus 2-minute Q&A) in class on a social media platform or a NLP research group of your choice.

First, please [sign up](#) to pick a date you want to present, and pick a social media platform (a list [here](#)) or a NLP research group (a list [here](#)).

Second, after your presentation in the class, [upload your slides](#) to OSU's Carmen system.



Memento



R 2000 · Crime film/Drama film · 1h 53m

8.5/10

[IMDb](#)

92%

[Rotten Tomatoes](#)

3/4

[Roger Ebert](#)

- We will take some photos next week!

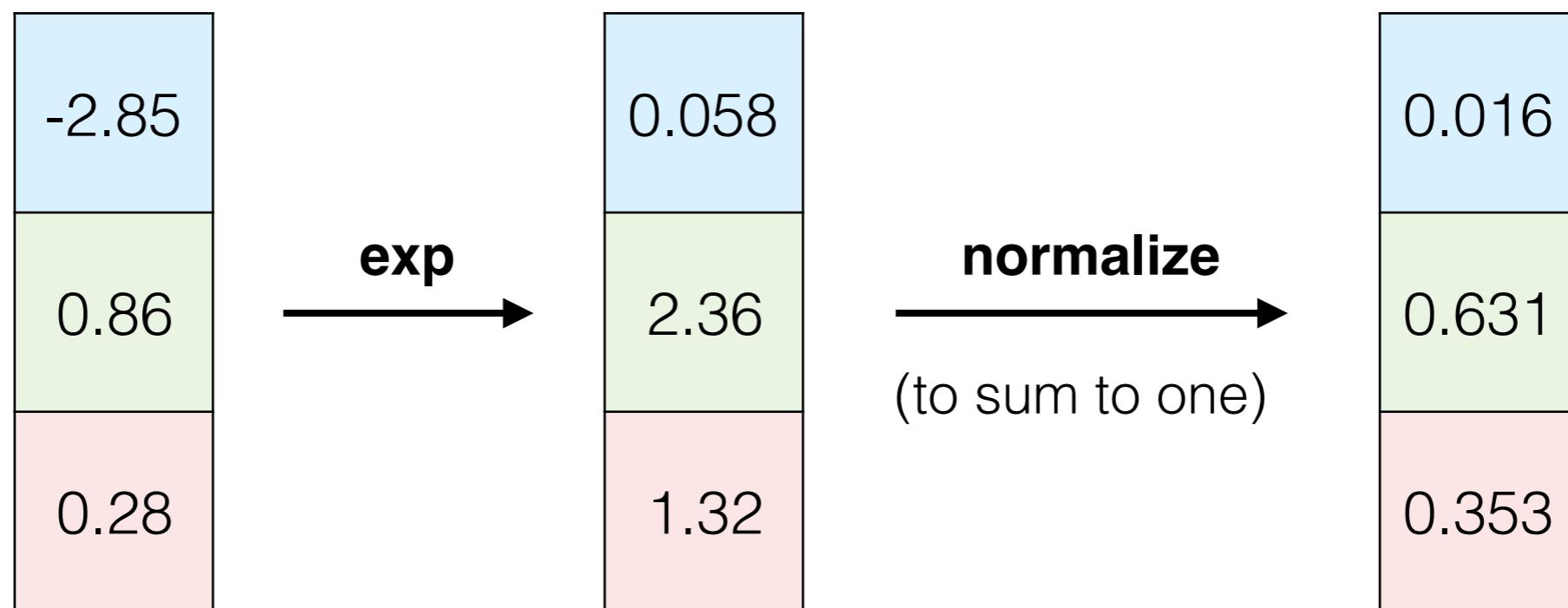
Quiz #2

- implement Softmax function in Python
(need to be computationally efficient)

```
1 import numpy as np
2
3 def softmax(x):
4
5     # Handle the special case: when vector x is only 1-dimensional
6     if x.ndim <= 1:
7         x = x - np.max(x)
8         ex = np.exp(x)
9         return ex / np.sum(ex)
10
11
12     ### YOUR CODE HERE -- for vectors that are not 1-dimensional
13
14
15
16
17
18
19     ### END YOUR CODE
20
21     return dist
22
23
24 # Check your Softmax implementation
25
26 print softmax(np.array([[101,102],[-1,-2]]))
```

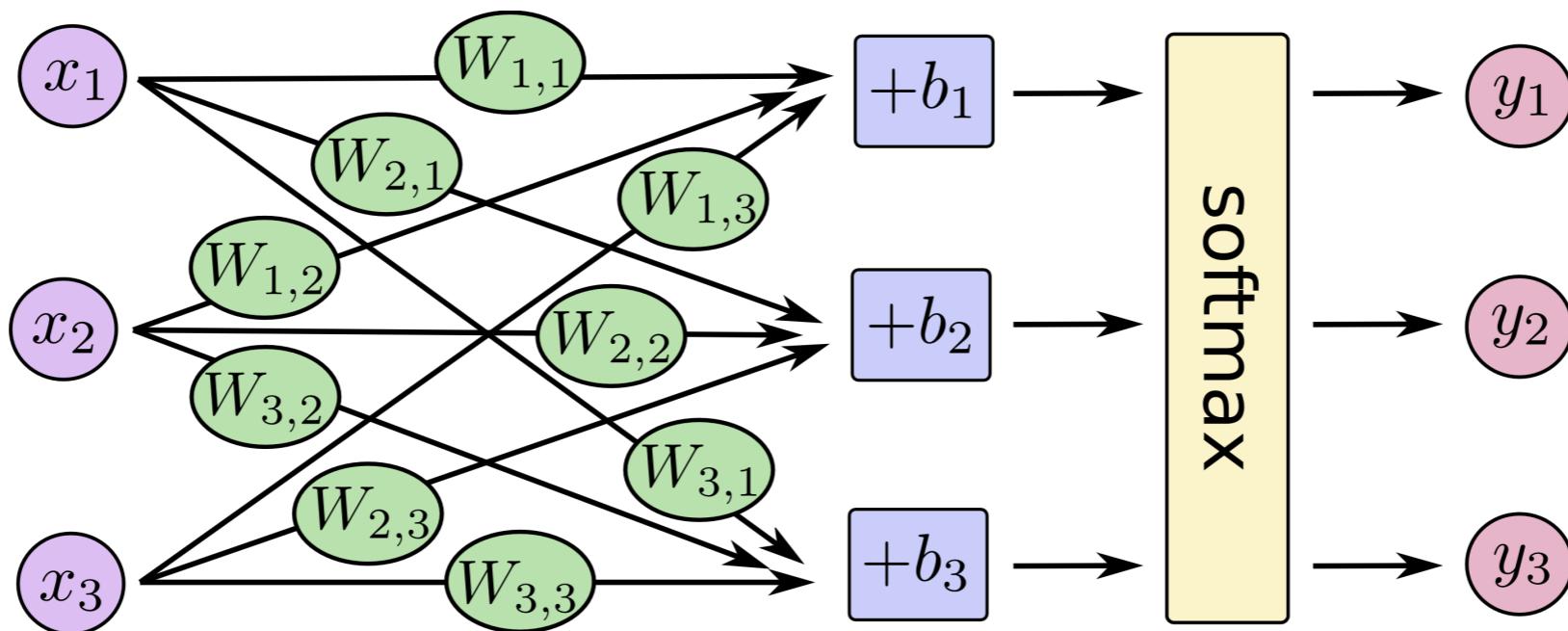
Softmax Function

$$\text{softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$$



Softmax Function

- softmax regression (multinomial logistic regression)
- often used as the output layer in neural networks



- We will learn later in the class

Quiz #2

- derivative of the Sigmoid function:
- use the chain rule:

if $f = g(u)$ and $u = h(x)$, i.e. $f(x) = g(h(x))$, then:

$$\frac{df}{dx} = \frac{df}{du} \frac{du}{dx} = \frac{dg(u)}{du} \frac{dh(x)}{dx}$$

Natural Language Processing

Dan Jurafsky



Language Technology

making good progress

mostly solved

Spam detection

Let's go to Agra!



Buy V1AGRA ...



Part-of-speech (POS) tagging

ADJ ADJ NOUN VERB ADV

Colorless green ideas sleep furiously.

Named entity recognition (NER)

PERSON ORG LOC

Einstein met with UN officials in Princeton

Natural Language Processing

Sentiment analysis

Best roast chicken in San Francisco!



The waiter ignored us for 20 minutes.



Coreference resolution

Carter told Mubarak he shouldn't run again.



Word sense disambiguation (WSD)

I need new batteries for my **mouse**.



Parsing



I can see Alcatraz from the window!

Machine translation (MT)

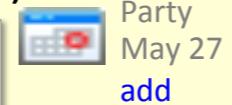
第13届上海国际电影节开幕...



The 13th Shanghai International Film Festival...

Information extraction (IE)

You're invited to our dinner party, Friday May 27 at 8:30



Party
May 27
add

still really hard

Question answering (QA)

Q. How effective is ibuprofen in reducing fever in patients with acute febrile illness?

Paraphrase

XYZ acquired ABC yesterday

ABC has been taken over by XYZ

Summarization

The Dow Jones is up

The S&P500 jumped

Housing prices rose



Economy is good

Dialog

Where is Citizen Kane playing in SF?



Castro Theatre at 7:30. Do you want a ticket?



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 Diffrent Social Media Sources?" IJCNLP 2013

Domain/Genre

- How different?

out-of-vocabulary
↓

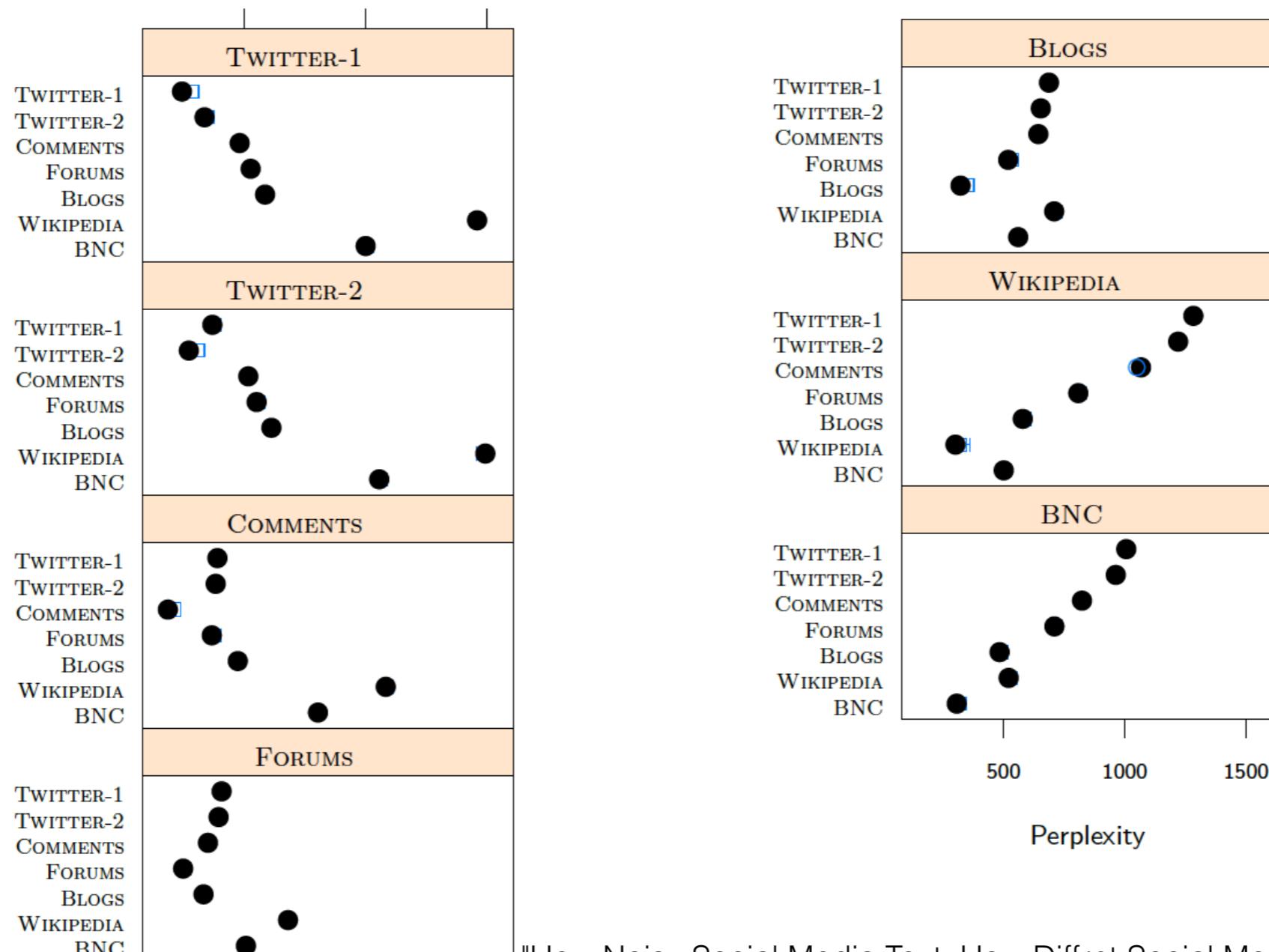
Corpus	Word length	Sentence length	%OOV
TWITTER-1	3.8 ± 2.4	9.2 ± 6.4	24.6
TWITTER-2	3.8 ± 2.4	9.0 ± 6.3	24.0
COMMENTS	3.9 ± 3.2	10.5 ± 10.1	19.8
FORUMS	3.8 ± 2.3	14.2 ± 12.7	18.1
BLOGS	4.1 ± 2.8	18.5 ± 24.8	20.6
WIKIPEDIA	4.5 ± 2.8	21.9 ± 16.2	19.0
BNC	4.3 ± 2.8	19.8 ± 14.5	16.9

Source: Baldwin et al.

"How Noisy Social Media Text, How Diffrent Social Media Sources?" IJCNLP 2013

Domain/Genre

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



Domain/Genre

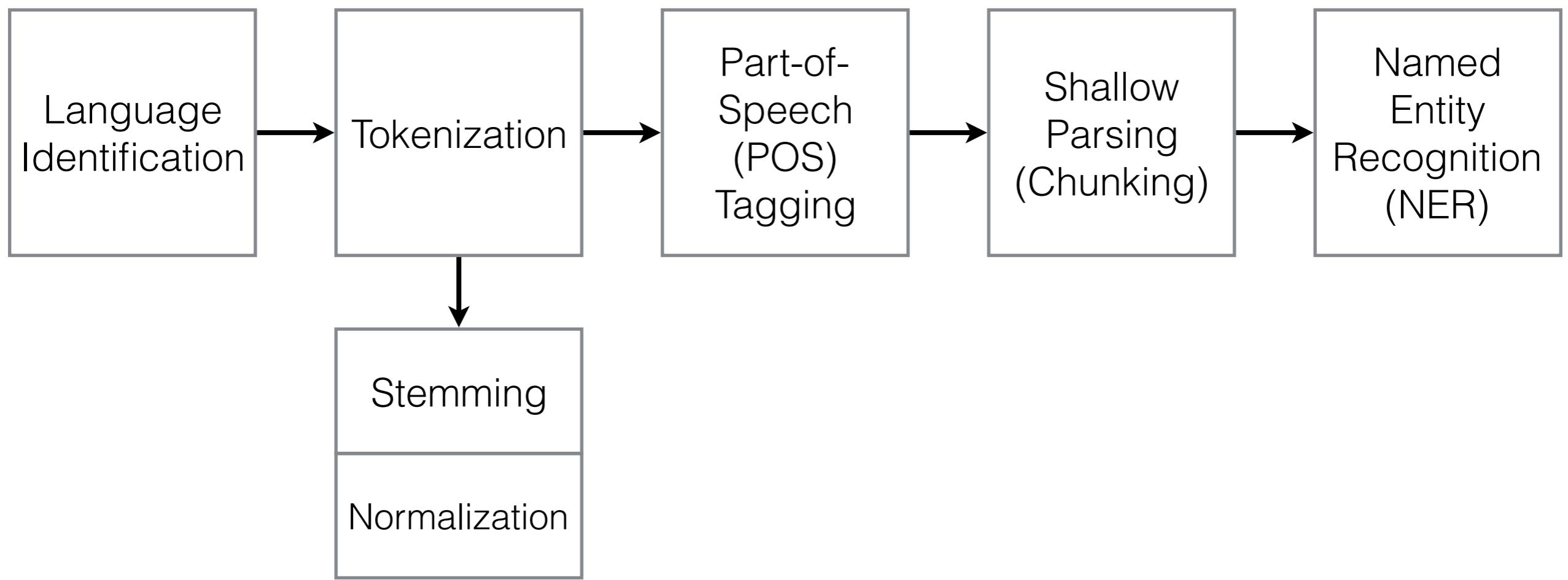
- What to do?
 - robust tools/models that works across domains
 - specific tools/models for Twitter data only — many techniques/algorithms are useful elsewhere
- (we will see examples of both in the class)

Domain/Genre

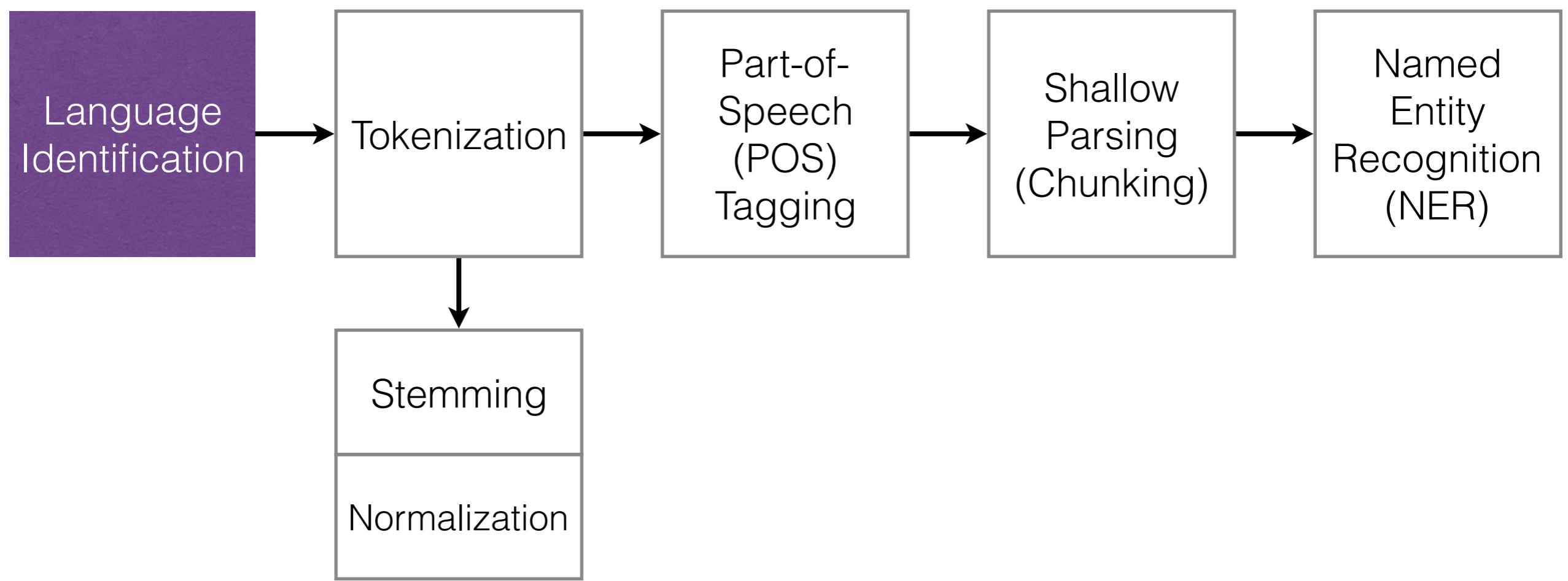
- Why so much Twitter?
 - publicly available (vs. SMS, emails)
 - large amount of data
 - large demand for research/commercial purpose
 - too different from well-edited text (which most NLP tools have been made for)

NLP Pipeline

NLP Pipeline



NLP Pipeline



Language Identification (a.k.a Language Detection)

 **Narendra Modi Hindi**
@narendramodiH

Follow

हर जगह छत्तीसगढ़ के लोगों से पूछा कि क्या कांग्रेस पर भरोसा किया जा सकता है और मुझे जवाब में एक शानदार नहीं मिला | nm4.in/1bsx4mV

 **Narendra Modi** 
@narendramodi

Follow

Отдаем большое значение организациям БРИКС и ШОС.
Надеюсь, что встречи в рамках саммитов будут продуктивными. @BRICS2015



 **Narendra Modi** 
@narendramodi

Follow

非常高兴再次与习近平主席会见。我们进行了全面的讨论，讨论了很多议题。@BRICS2015

 **Narendra Modi** 
@narendramodi

Follow

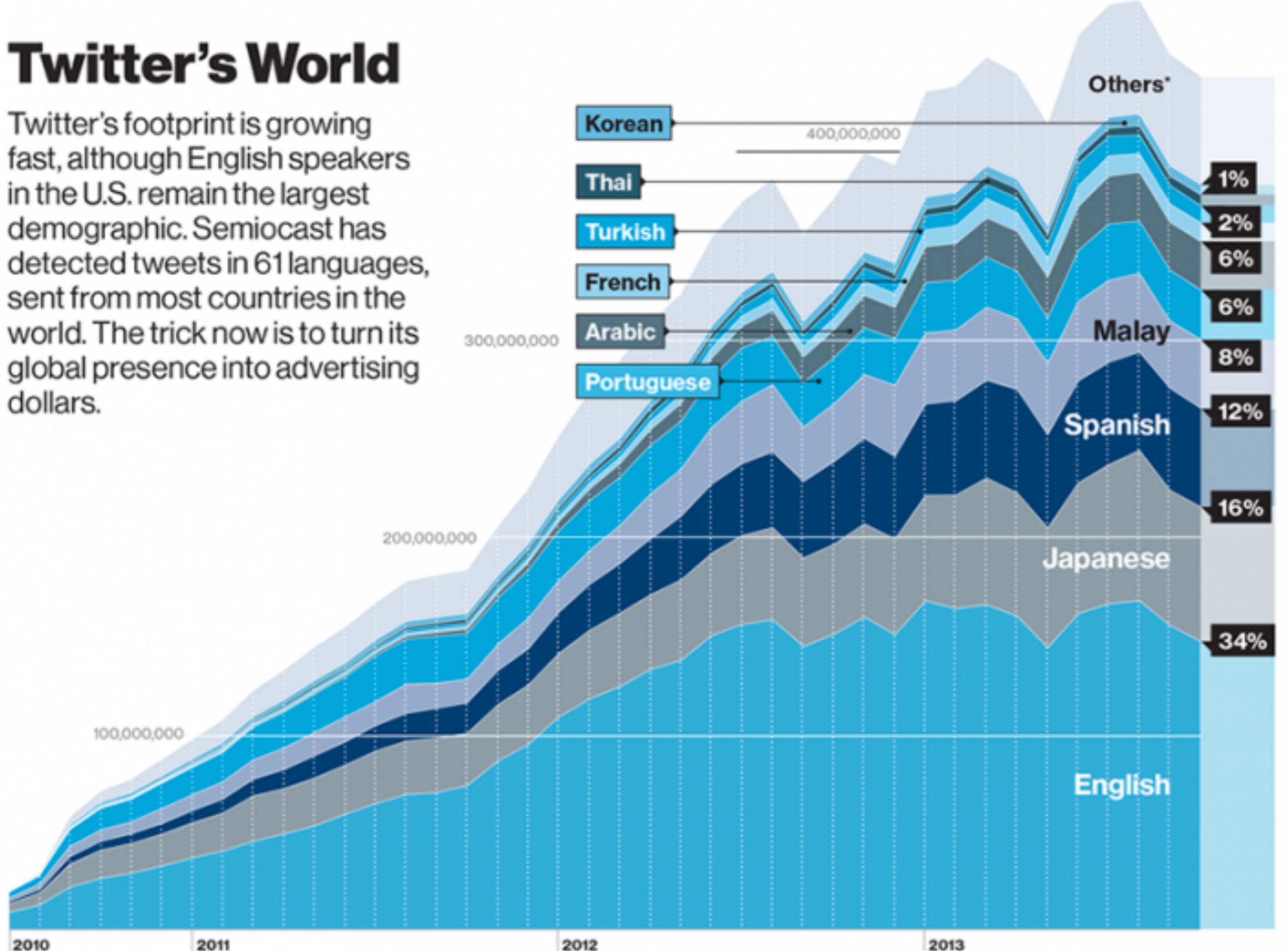
私は8月30日から日本を訪問する。印日関係を強化するこの訪問を、とても楽しみにしている。@AbeShinzo

LangID: why needed?

- Twitter is highly multilingual
- But NLP is often monolingual

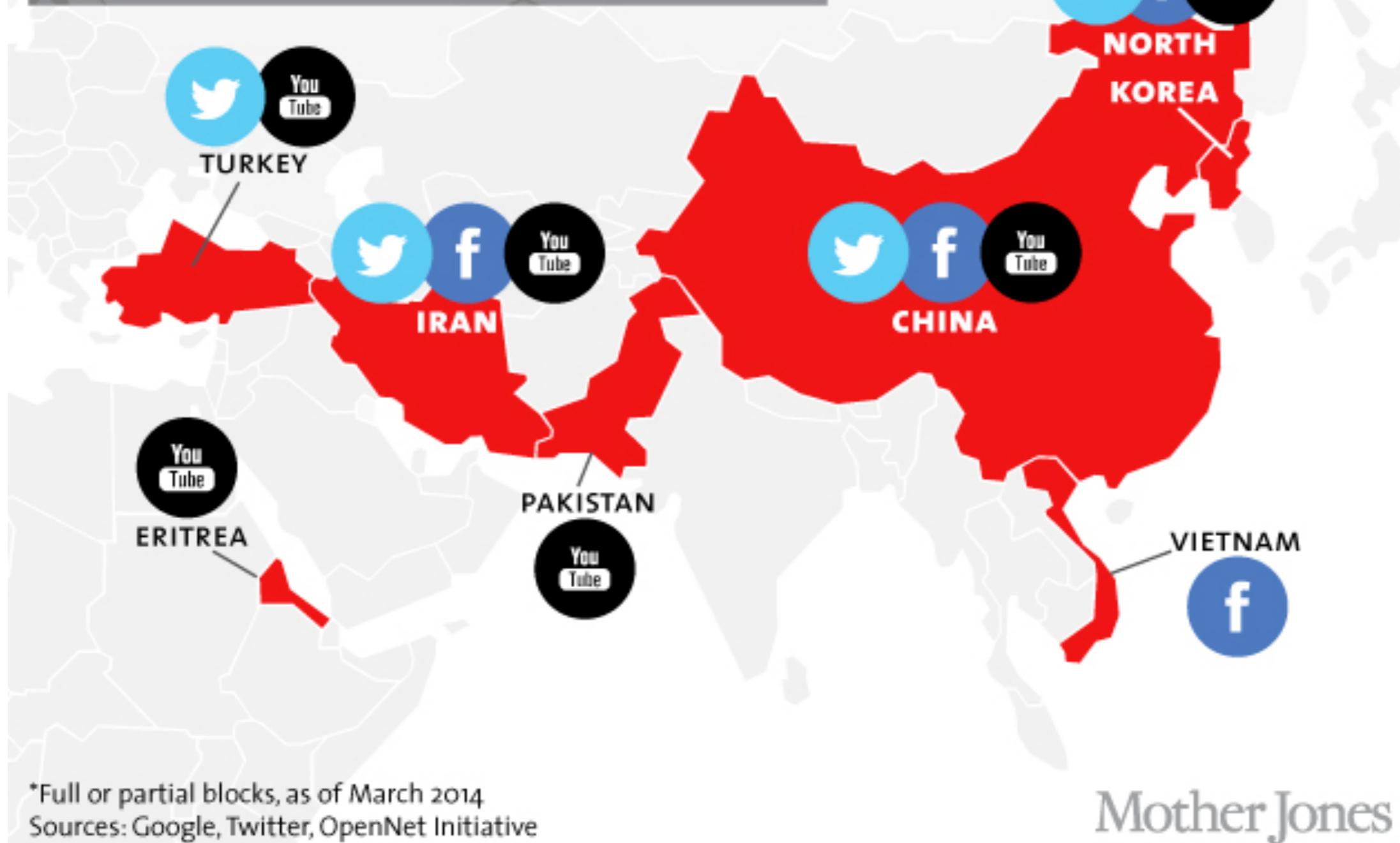
Twitter's World

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.



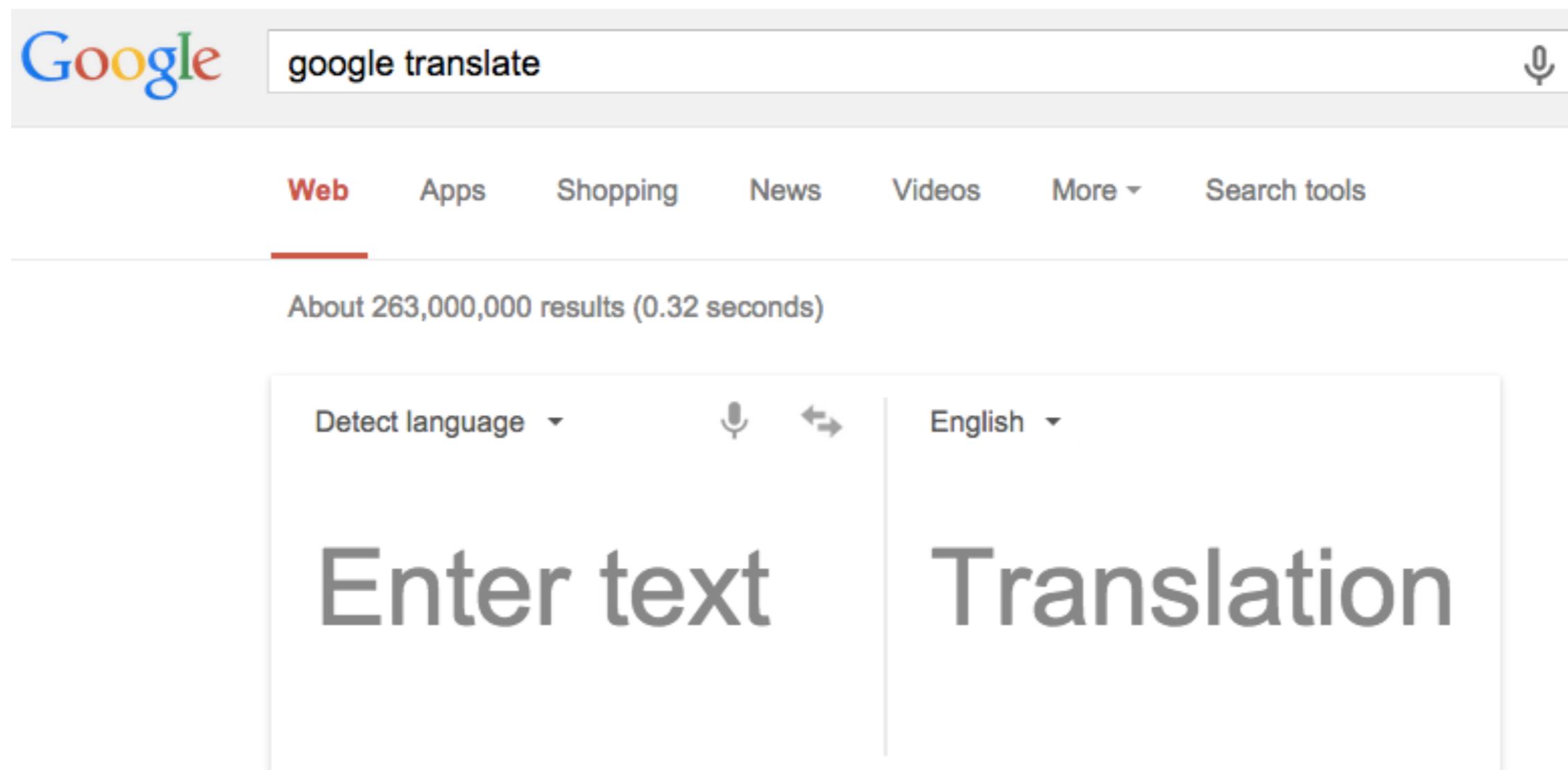
Social Media Under Fire

Countries that block Twitter, Facebook, or YouTube*



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

The screenshot shows the GitHub repository page for 'langid.py' by 'saffsd'. The page includes the repository URL, navigation tabs (This repository, Search, Pull requests, Issues, Gist), and repository statistics (225 commits, 5 branches, 0 releases, 3 contributors). A pull request list is displayed, showing merges from 'martinlh/master' and various commits to the 'master' branch.

Stand-alone language identification system

225 commits 5 branches 0 releases 3 contributors

Merge pull request #32 from martinlh/master ...

File	Description	Date
langid	Fixes ImportError on Python 3.	2 months ago
FEATURES	added a list of the 7480-feature model that is built into langid.py	a year ago
LICENSE	made license clearer	3 years ago
README.rst	made langid.py cross-compatible with Python2 and Python3	4 months ago
setup.cfg	restructure langid.py as python egg	3 years ago
setup.py	fixed issue #10 (and properly fixed #8)	2 years ago

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 \mathbf{d}
 - a fixed set of classes $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_j\}$
- Output:
 - a predicted class $\mathbf{c} \in \mathcal{C}$

Classification Method: Hand-crafted Rules

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

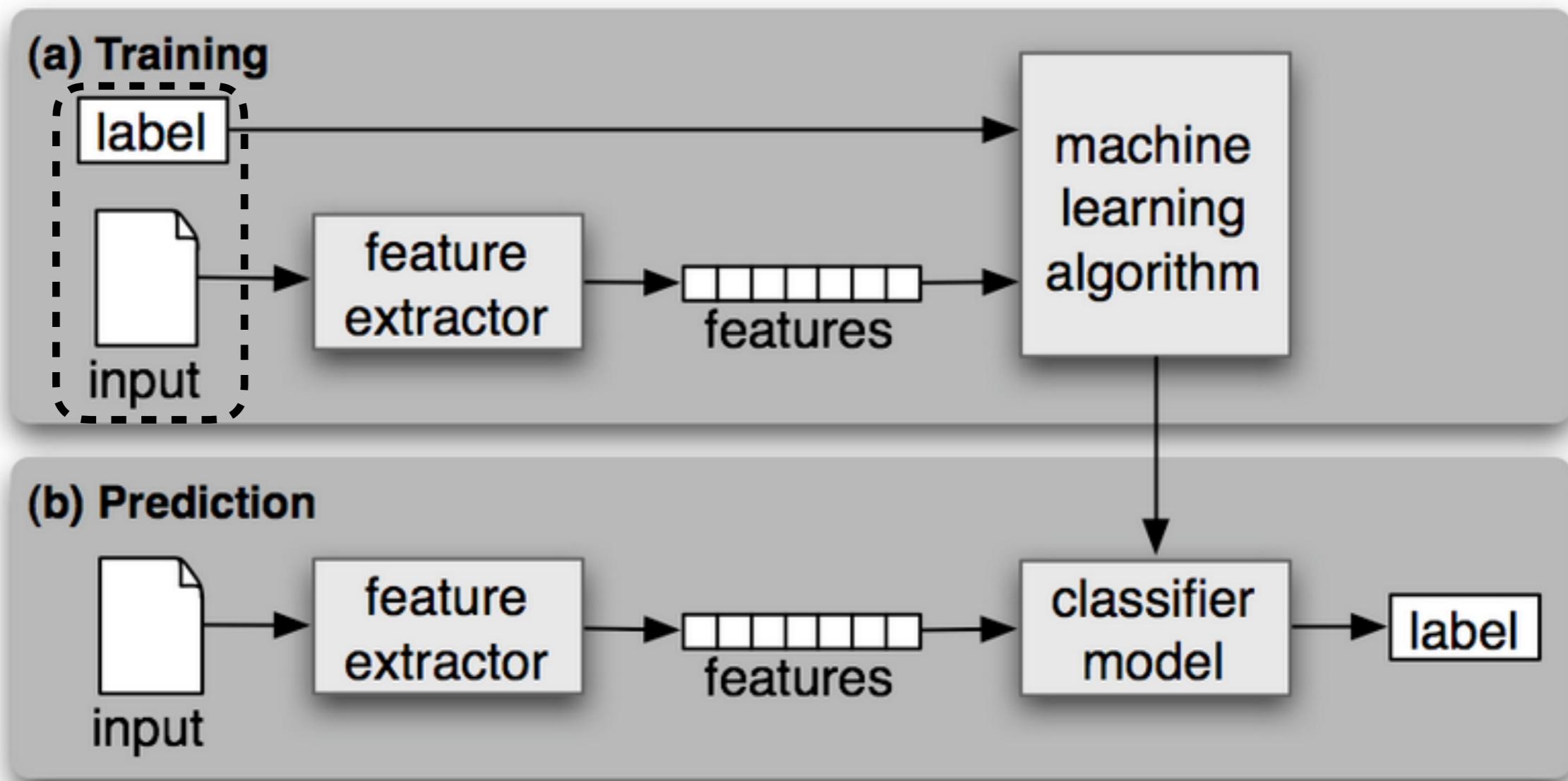
	English	Spanish
B		
banana		banana
banjo		banjo
bicycle		bicicleta
biography		biografía
blouse		blusa
brilliant		brillante

Classification Method:

Supervised Machine Learning

- Input:
 - a document \mathbf{d}
 - a fixed set of classes $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_j\}$
 - a training set of m hand-labeled documents
 $(\mathbf{d}_1, \mathbf{c}_1), \dots, (\mathbf{d}_m, \mathbf{c}_m)$
- Output:
 - a learned classifier $\gamma: \mathbf{d} \rightarrow \mathcal{C}$

Classification Method: Supervised Machine Learning



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

- a family of simple probabilistic classifiers based on Bayes' theorem with strong (naive) independence assumptions between the features.
- Bayes' Theorem:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

Naïve Bayes

- For a document \mathbf{d} , find the most probable class \mathbf{c} :

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



maximum a posteriori

Naïve Bayes

- For a document \mathbf{d} , find the most probable class \mathbf{c} :

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

$$= \arg \max_{c \in C} \frac{P(d \mid c)P(c)}{P(d)}$$

← Bayes Rule

Naïve Bayes

- For a document \mathbf{d} , find the most probable class \mathbf{c} :

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

$$= \arg \max_{c \in C} \frac{P(d \mid c)P(c)}{P(d)}$$

← Bayes Rule

$$= \arg \max_{c \in C} P(d \mid c)P(c)$$

← drop the denominator

Naïve Bayes

- document d represented as features t_1, t_2, \dots, t_n :

$$c_{MAP} = \arg \max_{c \in C} P(d | c)P(c)$$

$$= \arg \max_{c \in C} P(t_1, t_2, \dots, t_n | c)P(c)$$

Naïve Bayes

- document d represented as features t_1, t_2, \dots, t_n :

$$c_{MAP} = \arg \max_{c \in C} P(t_1, t_2, \dots, t_n | c) \underbrace{P(c)}_{\text{prior}}$$

how often
does this
class occur?
— simple count

Naïve Bayes

- document d represented as features t_1, t_2, \dots, t_n :

$$c_{MAP} = \arg \max_{c \in C} \frac{P(t_1, t_2, \dots, t_n | c) P(c)}{\text{likelihood} \quad \text{prior}}$$

$O(|T|^n \cdot |C|)$ parameters
 n = number of unique n-gram tokens

— need to make simplifying assumption

Naïve Bayes

- **Conditional Independence Assumption:**

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

$$P(t_1, t_2, \dots, t_n | c)$$

$$= P(t_1 | c) \cdot P(t_2 | c) \cdot \dots \cdot P(t_n | c)$$

Naïve Bayes

- For a document \mathbf{d} , find the most probable class \mathbf{c} :

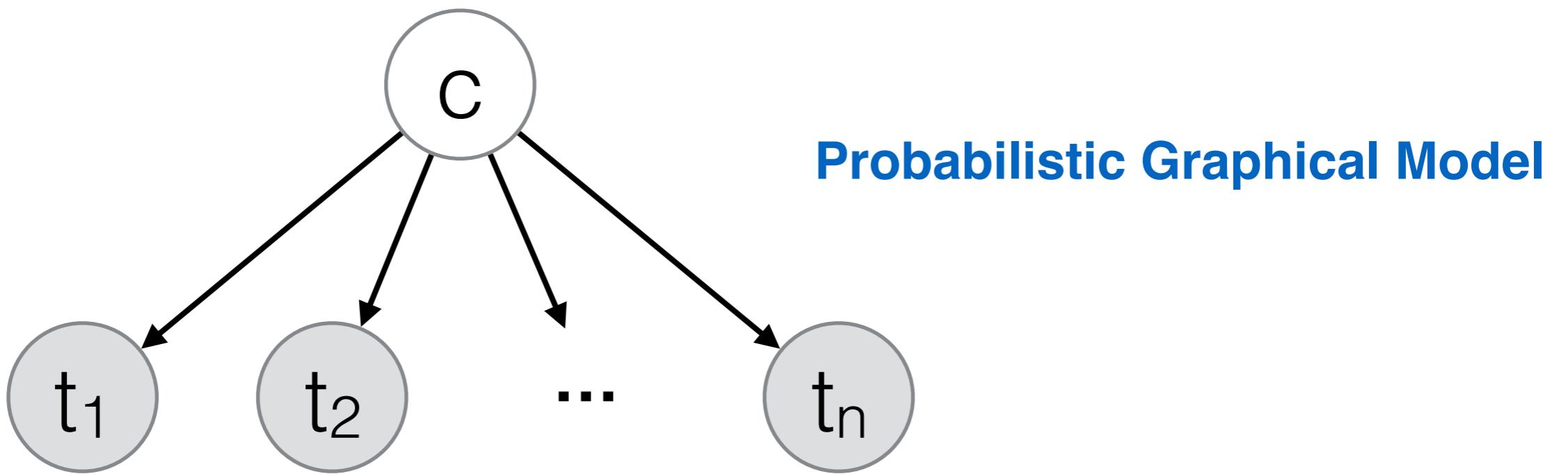
$$c_{MAP} = \arg \max_{c \in C} P(t_1, t_2, \dots, t_n \mid c)P(c)$$



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

Naïve Bayes

$$c_{NB} = \arg \max_{c \in C} P(c) \prod_{t_i \in d} P(t_i | c)$$



Variations of Naïve Bayes

$$c_{MAP} = \arg \max_{c \in C} P(d | c)P(c)$$

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

Variations of Naïve Bayes

$$c_{MAP} = \arg \max_{c \in C} P(d | c)P(c)$$

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

LangID features

English

- n-grams features:
 - 1-gram:
“the” “following” “Wikipedia”
“en” “español” ...
 - 2-gram:
“the following” “following is”
“Wikipedia en” “en español” ...
 - 3-gram:
....

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

$$c_{NB} = \arg \max_{c \in C} P(c) \prod_{t_i \in d} P(t_i | c)$$

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

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

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

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	?

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

$$P(\text{en})=3/4 \quad P(\text{sp})=1/4$$

$$P(\text{"Wikipedia"} | \text{en}) = 3/8, P(\text{"Wikipedia"} | \text{es}) = 1/3$$

$$P(\text{"español"} | \text{en}) = 0/8, P(\text{"español"} | \text{es}) = 1/3$$

$$P(\text{"el"} | \text{en}) = 0/8, P(\text{"el"} | \text{es}) = 0/3$$

$$P(\text{en} | \text{doc5}) = 3/4 \times 3/8 \times 0/8 \times 0/8 = 0$$

$$P(\text{es} | \text{doc5}) = 1/4 \times 2/9 \times 1/3 \times 0/3 = 0$$

Naïve Bayes

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

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

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

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

smooth
→

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

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	?

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

$$P(\text{en})=3/4 \quad P(\text{sp})=1/4$$

$$P(\text{"Wikipedia"} | \text{en}) = 3+1/8+6, \quad P(\text{"Wikipedia"} | \text{sp}) = 1+1/3+6$$

$$P(\text{"español"} | \text{en}) = 0+1/8+6, \quad P(\text{"español"} | \text{sp}) = 1+1/3+6$$

$$P(\text{"el"} | \text{en}) = 0+1/8+6, \quad P(\text{"el"} | \text{sp}) = 0+1/3+6$$

$$P(\text{en} | \text{doc5}) = 3/4 \times 4/14 \times 1/14 \times 1/14 = 0.00109$$

$$P(\text{sp} | \text{doc5}) = 1/4 \times 2/9 \times 2/9 \times 1/9 = 0.00137$$

Naïve Bayes

- **Pros: (works well for spam filtering, text classification, sentiment analysis, language identification)**
 - simple (no iterative learning)
 - fast and light-weighted
 - less parameters, so need less training data
 - even if the NB assumption doesn't hold, a NB classifier still often performs surprisingly well in practice
- **Cons**
 - assumes independence of features
 - can't model dependencies/structures

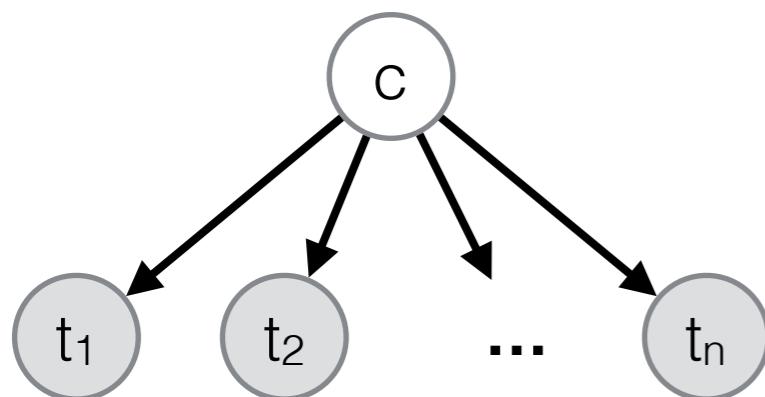
Correlated Features

- For example, for spam email classification, word “win” often occurs together with “free”, “prize”.
- Solution:
 - feature selection
 - or other models (**e.g. logistic/softmax regression**)

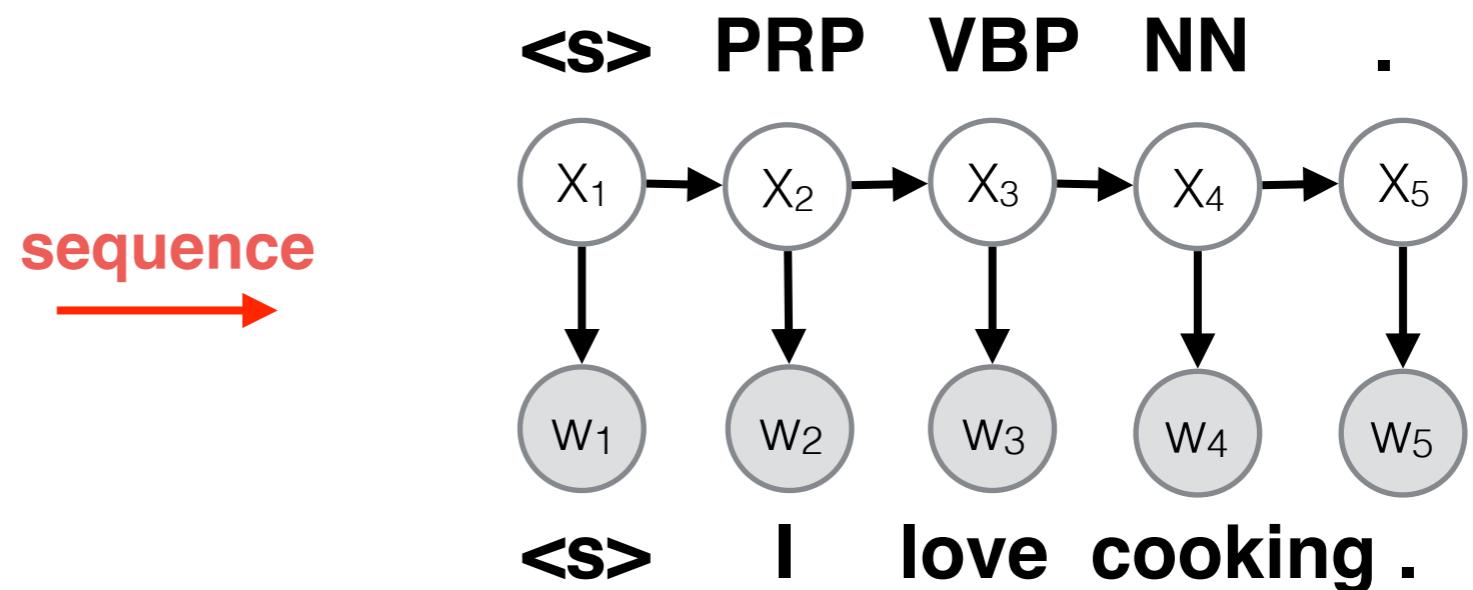
Model Structure

- For example, the word order matters in part-of-speech tagging:

Naive Bayes



Hidden Markov Model (HMM)



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

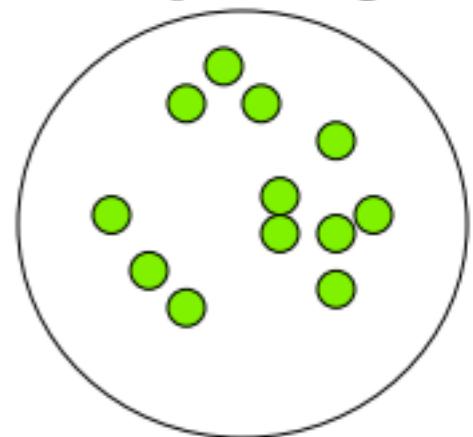
- main techniques:
 - **Multinomial Naïve Bayes**
 - diverse training data from multiple domains (Wikipedia, Reuters, Debian, etc.)
 - plus **feature selection** using **Information Gain (IG)** to choose features that are informative about language, but not informative about domain

Information Gain

- Information Gain:

$$IG(Y | X) = H(Y) - H(Y | X)$$

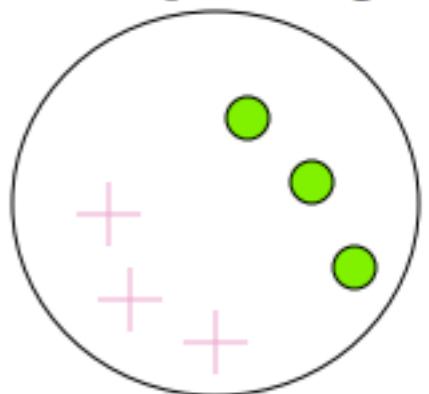
$H(X) = 0$
Minimum impurity



- Entropy:

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

$H(X) = 1$
Maximum impurity



Information Gain

wealth values: poor rich

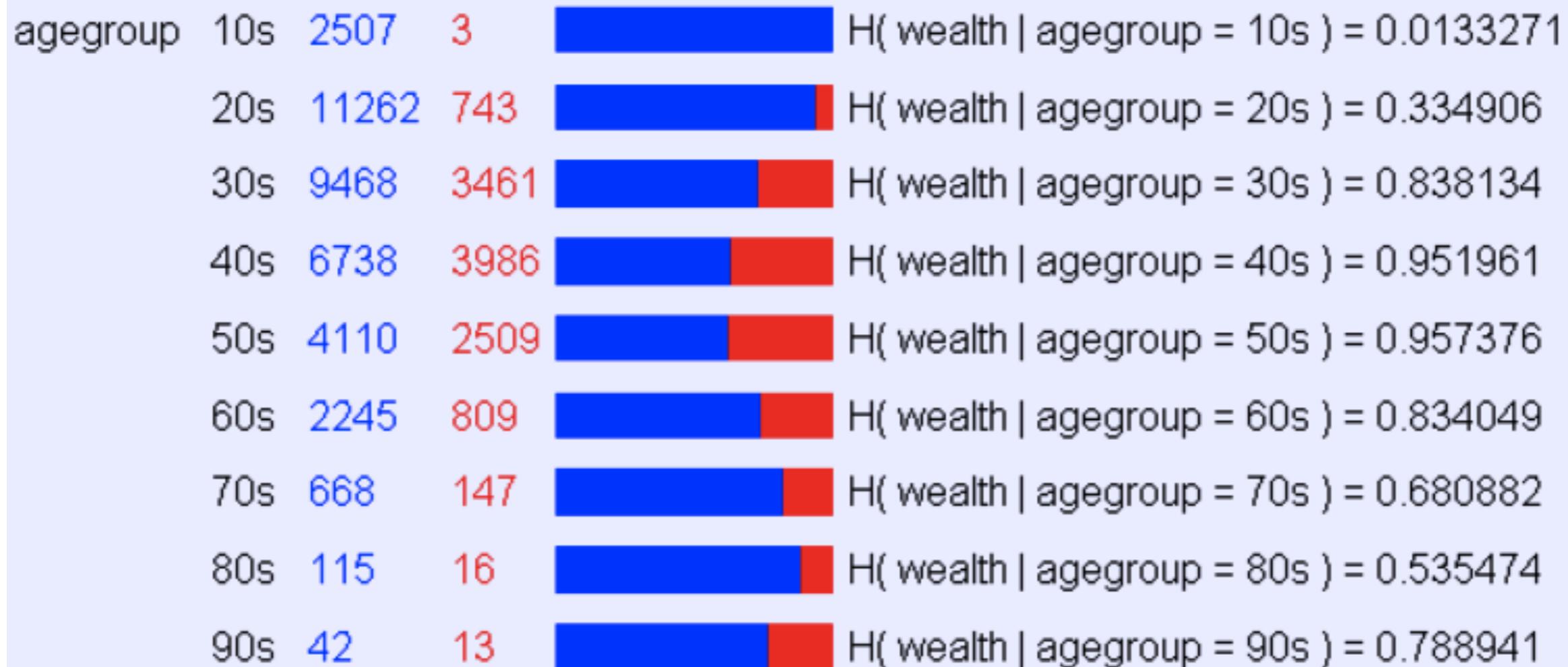
gender	Female	14423	1769		$H(\text{wealth} \text{gender} = \text{Female}) = 0.497654$
	Male	22732	9918		$H(\text{wealth} \text{gender} = \text{Male}) = 0.885847$

$H(\text{wealth}) = 0.793844$ $H(\text{wealth}|\text{gender}) = 0.757154$

$\text{IG}(\text{wealth}|\text{gender}) = 0.0366896$

Information Gain

wealth values: poor rich



$$H(\text{wealth}) = 0.793844 \quad H(\text{wealth}|\text{agegroup}) = 0.709463$$

$$\text{IG}(\text{wealth}|\text{agegroup}) = 0.0843813$$

Information Gain used for?

- choose features that are informative (most useful) for discriminating between the classes.

Wealth

$$IG(\text{wealth} | \text{gender}) = 0.0366896$$

$$IG(\text{wealth} | \text{agegroup}) = 0.0843813$$

Longevity

$$IG(\text{LongLife} | \text{HairColor}) = 0.01$$

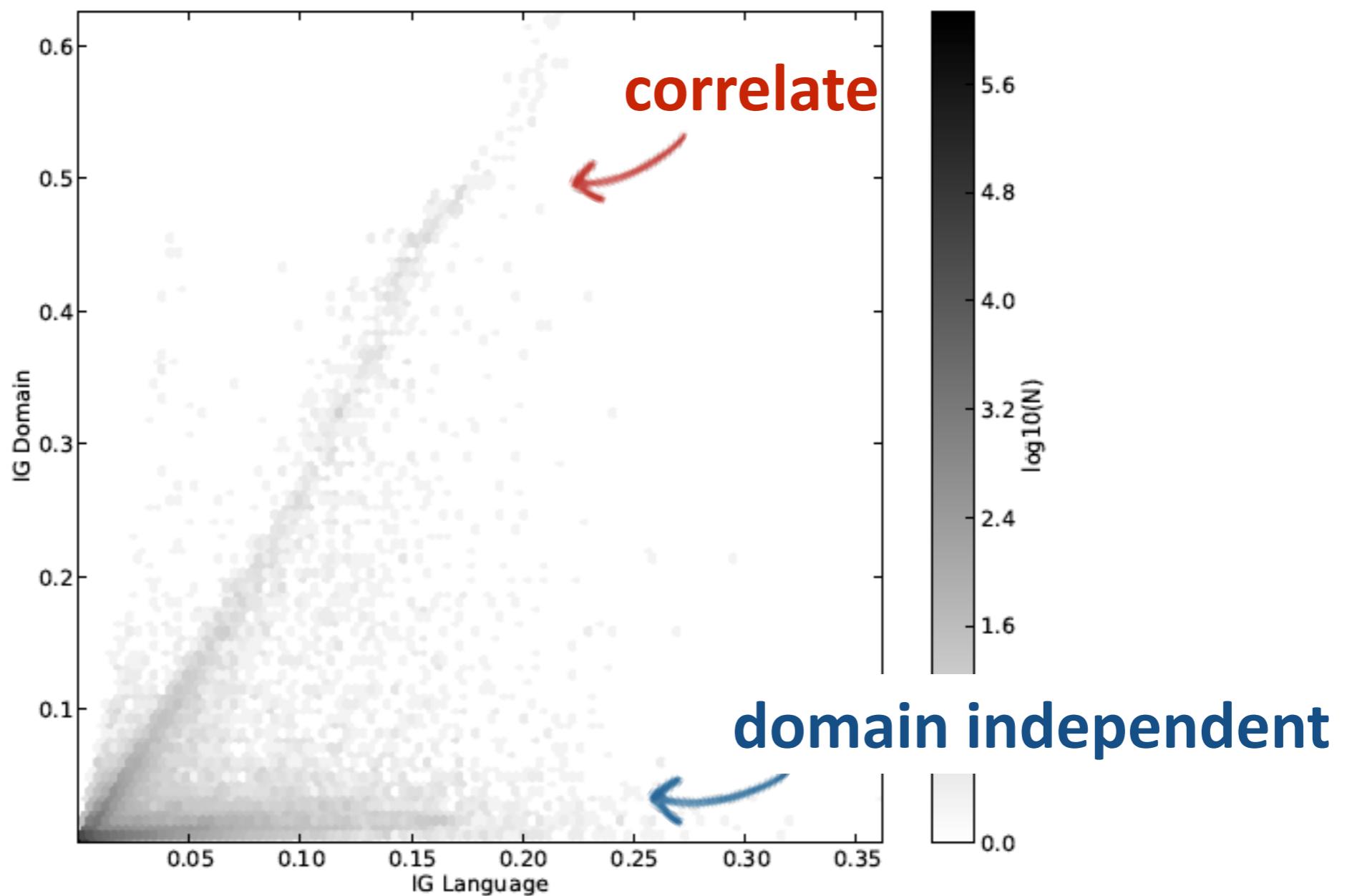
$$IG(\text{LongLife} | \text{Smoker}) = 0.2$$

$$IG(\text{LongLife} | \text{Gender}) = 0.25$$

$$IG(\text{LongLife} | \text{LastDigitOfSSN}) = 0.00001$$

LangID Tool: langid.py

- feature selection using Information Gain (IG)

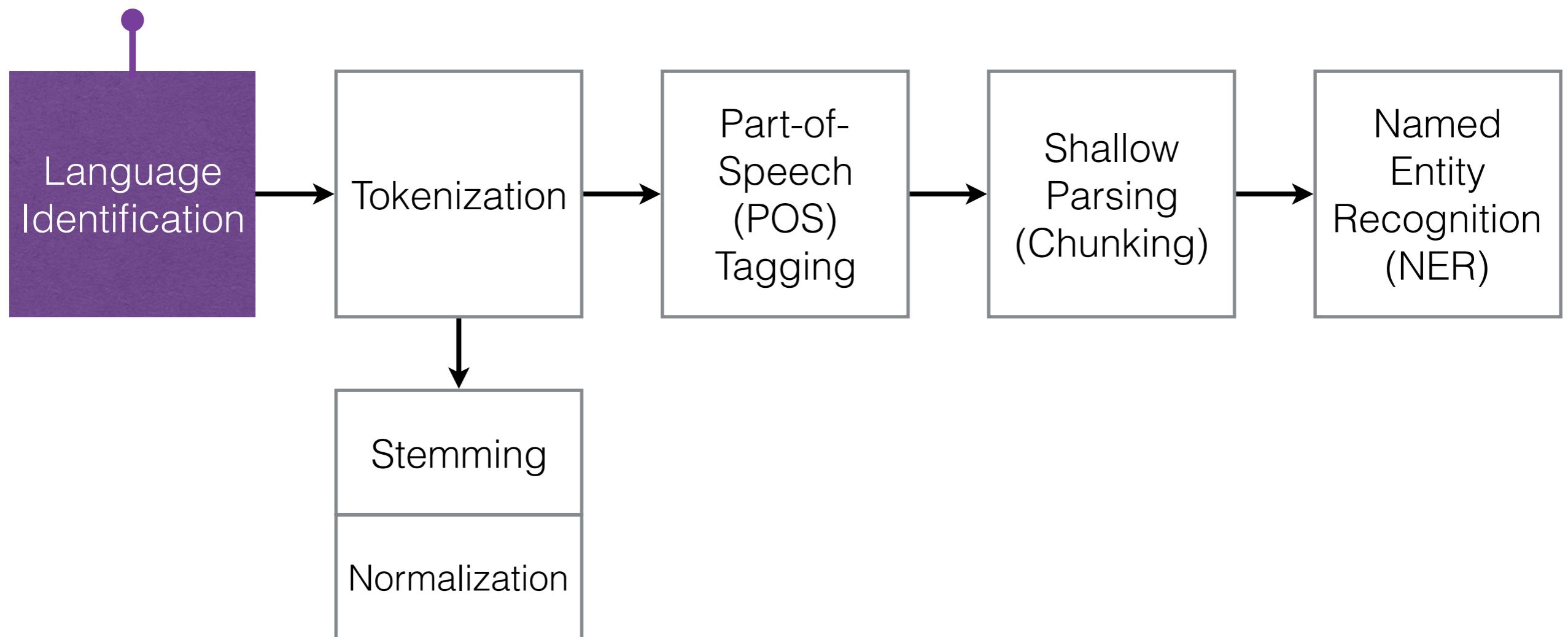


LangID Tool: langid.py

- 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

Summary

classification (Naïve Bayes)



Sign up for
in-class presentation
(by next week)

socialmedia-class.org