Social Media & Text Analysis

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

CSE 5539-0010 Ohio State University
Instructor: Alan Ritter
Website: socialmedia-class.org

In-class Presentation

- a 10-minute presentation plus 2-minute Q&A (20 points)
  - A Social Media Platform or a NLP Researcher
  - 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 Researcher : In-class Presentation (20 points)

You will pair together (2 students) and give a 10-minute presentation (plus 2-minute Q&A) in class about a social media platform (an incomplete list here) and a NLP researcher of your choice (an incomplete list of NLP groups here). You are also encouraged to find other NLP researchers that are not on this list through CS department homepages or top NLP conferences/jounals (e.g. ACL, NAACL, TACL, EMNLP).

Reading #1

What is Twitter, a Social Network or a News Media?

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon
Department of Computer Science, KAIST
335 Gwahangno, Yuseong-gu, Daejeon, Korea
{haewoon, chlee, hosung}@an.kaist.ac.kr, sbmoon@kaist.edu

ABSTRACT

Twitter, a microblogging service less than three years old, commands more than 41 million users as of July 2009 and is growing fast. Twitter users tweet about any topic within the 140-character limit and follow others to receive their tweets. The goal of this paper is to study the topological characteristics of Twitter and its power as a new medium of information sharing.

We have crawled the entire Twitter site and obtained 41.7 million user profiles, 1.47 billion social relations, 4,262 trending topics, and 106 million tweets. In its follower-following topology analysis we have found a non-power-law follower distribution, a short effective diameter, and low reciprocity, which all mark a deviation from known characteristics of human social networks [28]. In order to identify influentials on Twitter, we have ranked users by the number of followers and by PageRank and found two rankings to be similar. Ranking by retweets differs from the previous two rankings, indicating a gap in influence inferred from the number of followers and that from the popularity of one's tweets. We have analyzed the

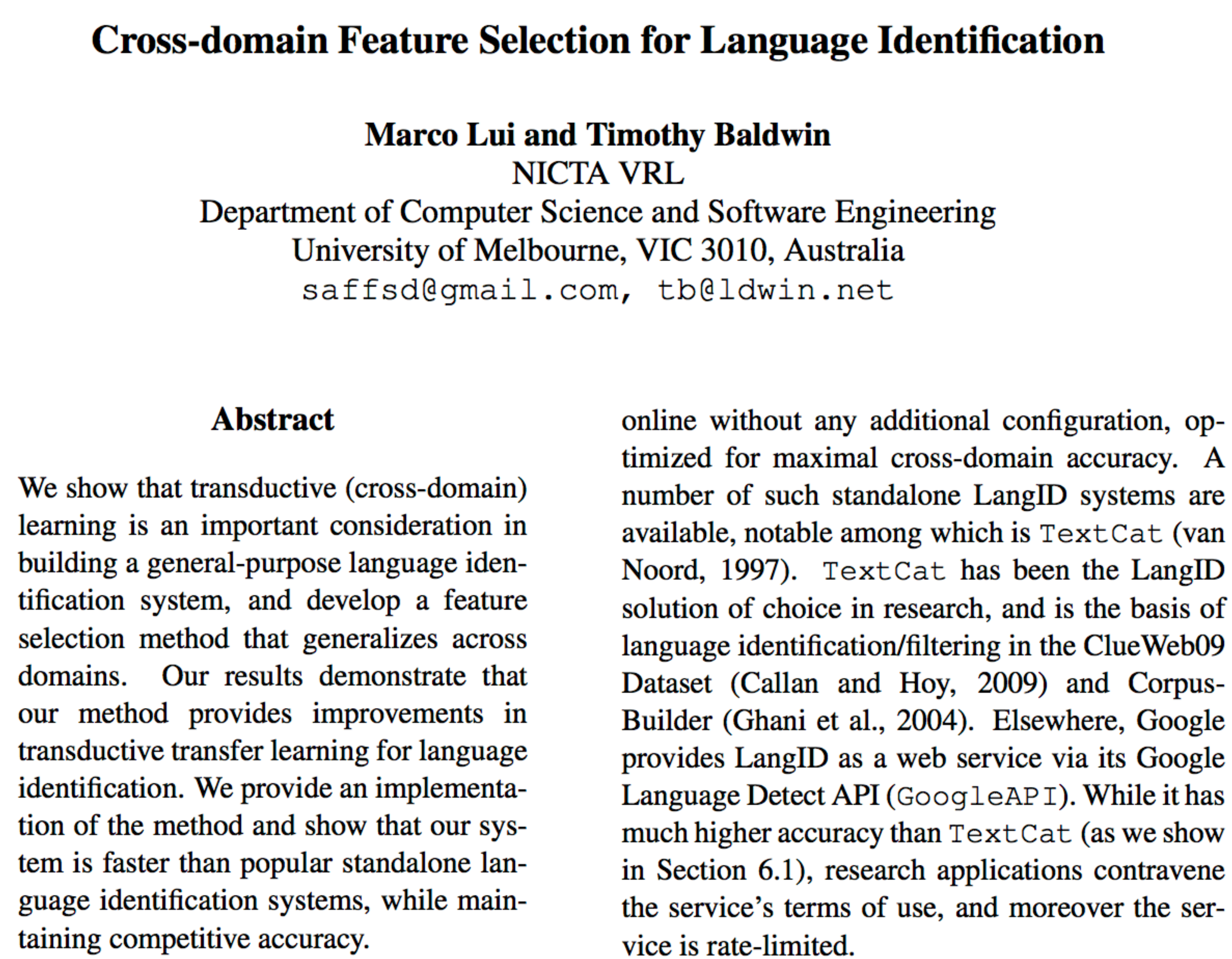
1. INTRODUCTION

Twitter, a microblogging service, has emerged as a new medium in spotlight through recent happenings, such as an American student jailed in Egypt and the US Airways plane crash on the Hudson river. Twitter users follow others or are followed. Unlike on most online social networking sites, such as Facebook or MySpace, the relationship of following and being followed requires no reciprocation. A user can follow any other user, and the user being followed need not follow back. Being a follower on Twitter means that the user receives all the messages (called *tweets*) from those the user follows. Common practice of responding to a tweet has evolved into well-defined markup culture: RT stands for retweet, '@' followed by a user identifier address the user, and '#' followed by a word represents a hashtag. This well-defined markup vocabulary combined with a strict limit of 140 characters per posting conveniences users with brevity in expression. The *retweet* mechanism empowers users to spread information of their choice beyond the reach of the original tweet's followers.

Reading #1

What is Twitter, a social network or a news media? - ACM Digital Library
dl.acm.org/citation.cfm?id=1772751
by H Kwak - 2010 - Cited by 5361 - Related articles
Apr 26, 2010 - **Twitter**, a microblogging service less than three years old, commands more than 41 million users as of July 2009 and is growing fast. **Twitter** ...
Abstract · Authors · References · Cited By

Reading #2


Cross-domain Feature Selection for Language Identification

Marco Lui and Timothy Baldwin
NICTA VRL
Department of Computer Science and Software Engineering
University of Melbourne, VIC 3010, Australia
saffsd@gmail.com, tb@ldwin.net

Abstract

We show that transductive (cross-domain) learning is an important consideration in building a general-purpose language identification system, and develop a feature selection method that generalizes across domains. Our results demonstrate that our method provides improvements in transductive transfer learning for language identification. We provide an implementation of the method and show that our system is faster than popular standalone language identification systems, while maintaining competitive accuracy.

online without any additional configuration, optimized for maximal cross-domain accuracy. A number of such standalone LangID systems are available, notable among which is TextCat (van Noord, 1997). TextCat has been the LangID solution of choice in research, and is the basis of language identification/filtering in the ClueWeb09 Dataset (Callan and Hoy, 2009) and CorpusBuilder (Ghani et al., 2004). Elsewhere, Google provides LangID as a web service via its Google Language Detect API (GoogleAPI). While it has much higher accuracy than TextCat (as we show in Section 6.1), research applications contravene the service's terms of use, and moreover the service is rate-limited.

Natural Language Processing

Dan Jurafsky

Language Technology

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

making good progress

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 Diffrnt Social Media Sources?" IJCNLP 2013

Domain/Genre

- How different?

out-of-vocabulary

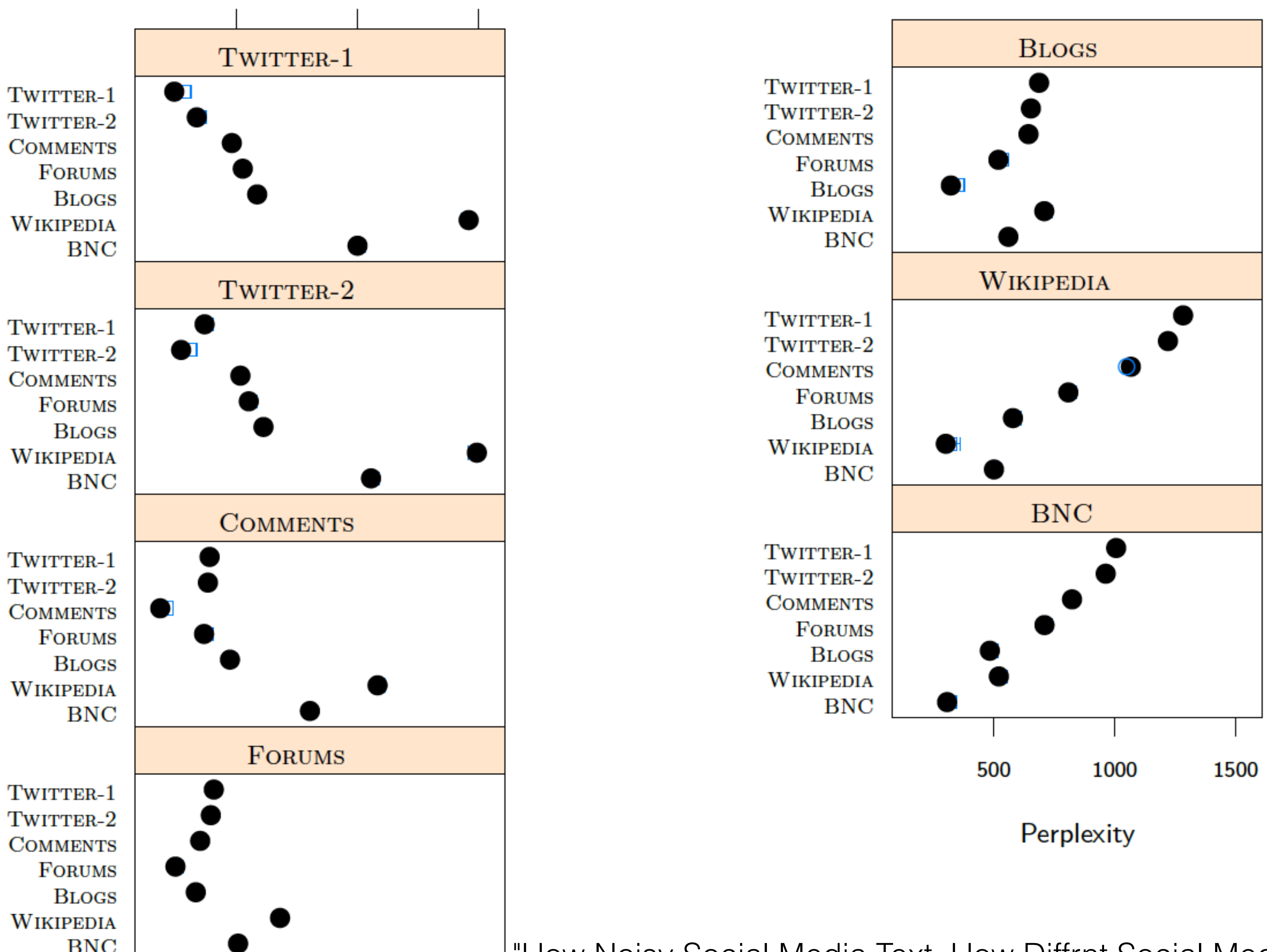
Corpus	Word length	Sentence length	%OOV
TWITTER-1	3.8±2.4	9.2±6.4	**24.6**
TWITTER-2	3.8±2.4	9.0±6.3	**24.0**
COMMENTS	3.9±3.2	10.5±10.1	**19.8**
FORUMS	3.8±2.3	14.2±12.7	**18.1**
BLOGS	4.1±2.8	18.5±24.8	**20.6**
WIKIPEDIA	4.5±2.8	21.9±16.2	**19.0**
BNC	4.3±2.8	19.8±14.5	**16.9**

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

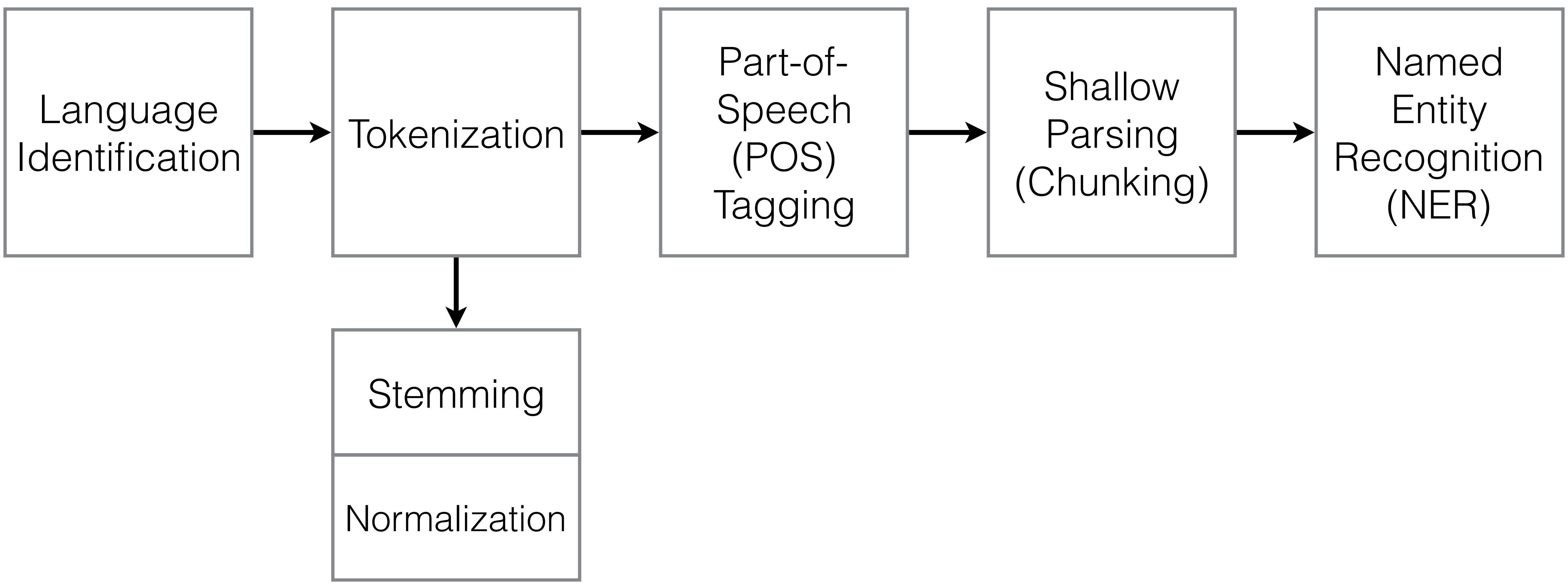
(we will see examples of both in the class)

Domain/Genre

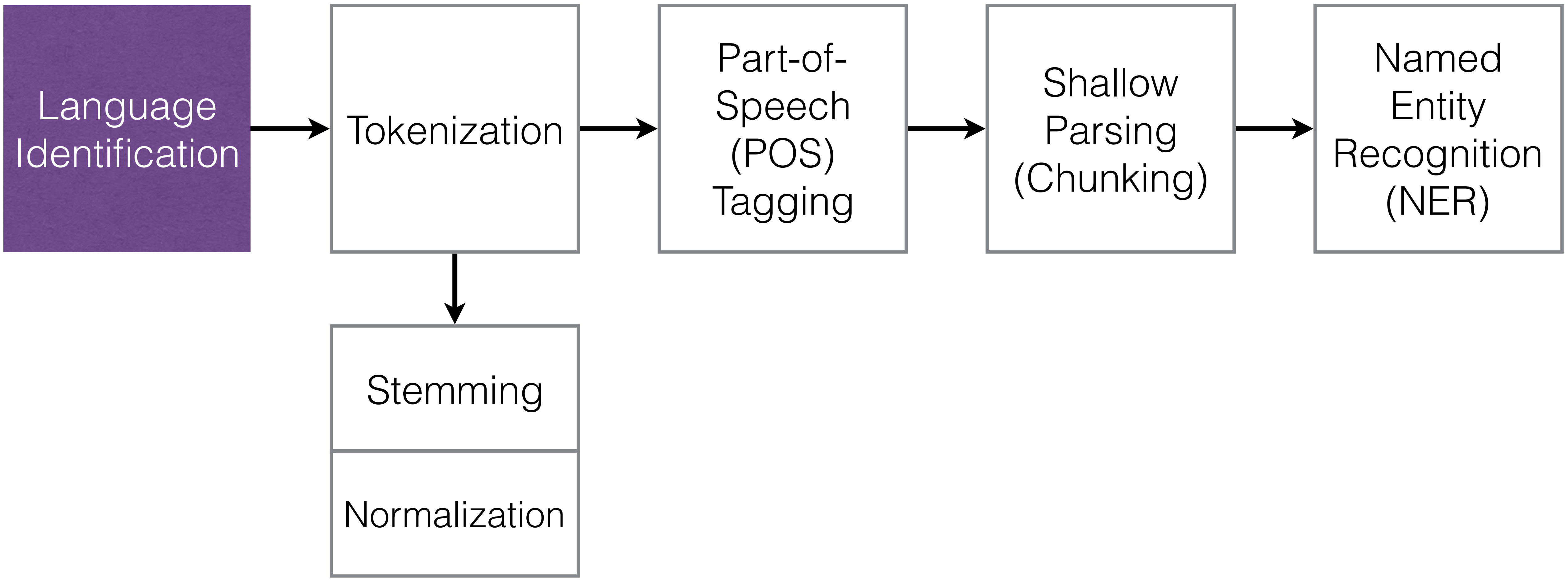
- Why so much Twitter?
  - publicly available (vs. SMS, emails)
  - large amount of data
  - large demand for research/commercial purpose
  - too different from well-edited text (which most NLP tools have been made for)

NLP Pipeline

NLP Pipeline

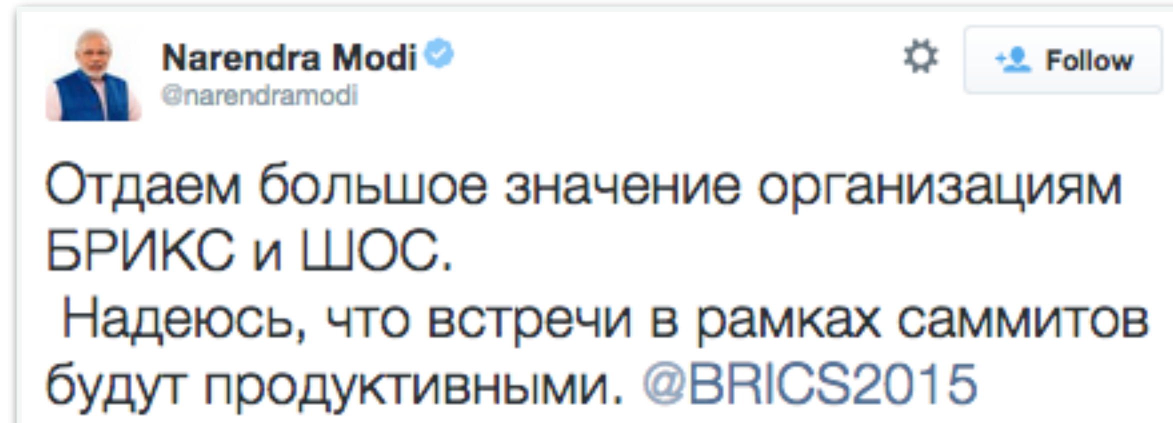


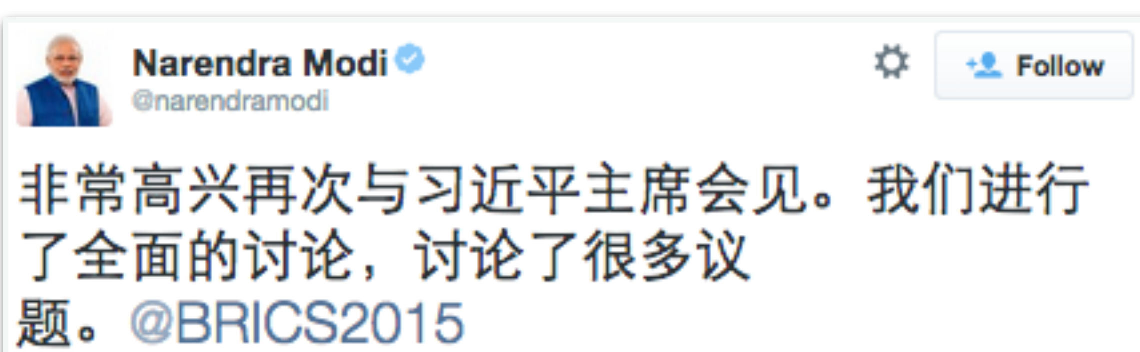
NLP Pipeline

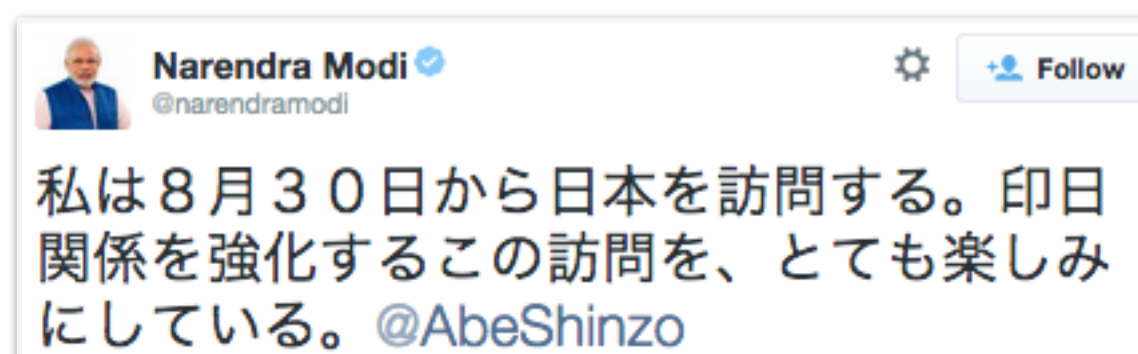


Language Identification

(a.k.a Language Detection)





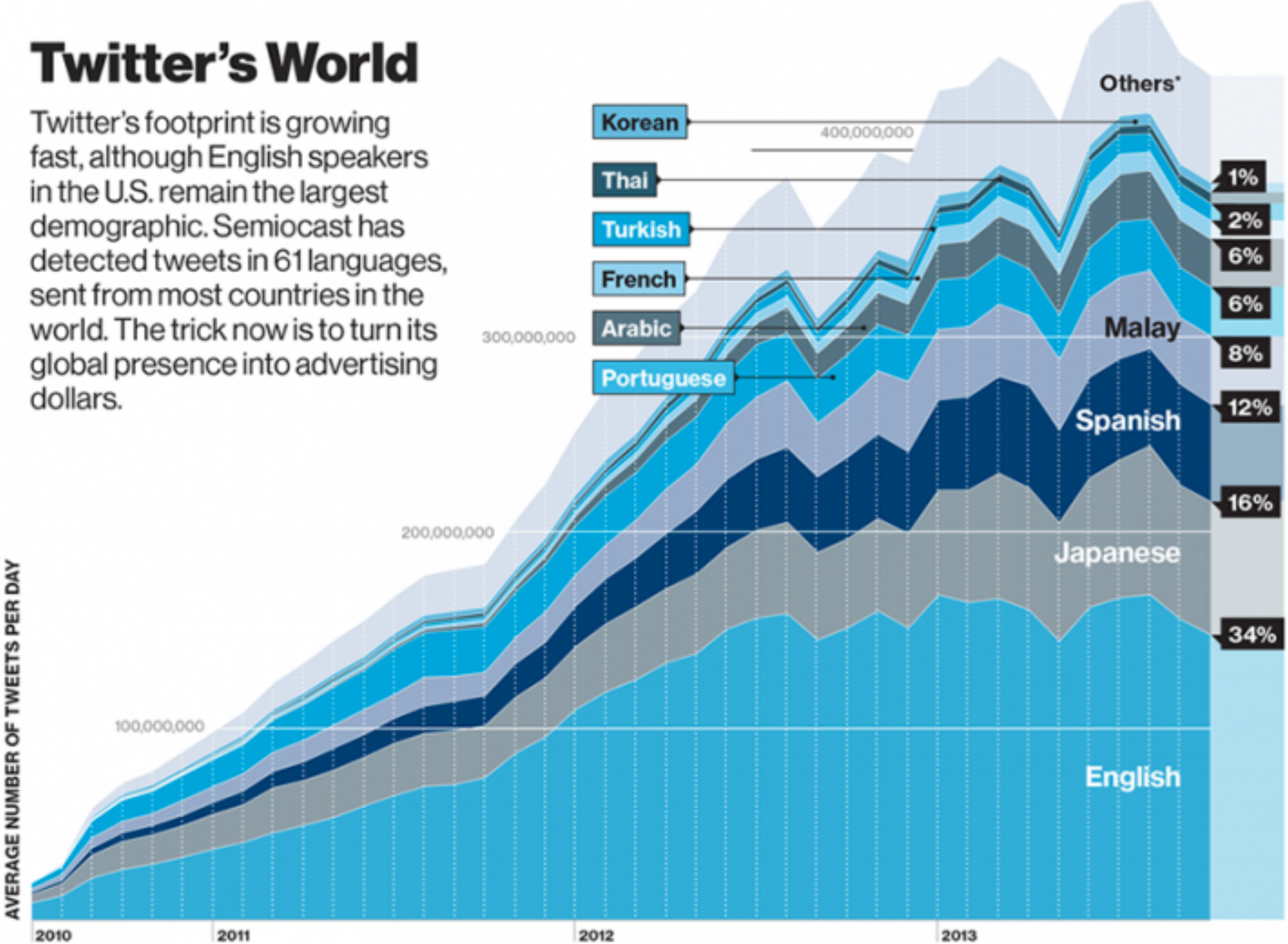


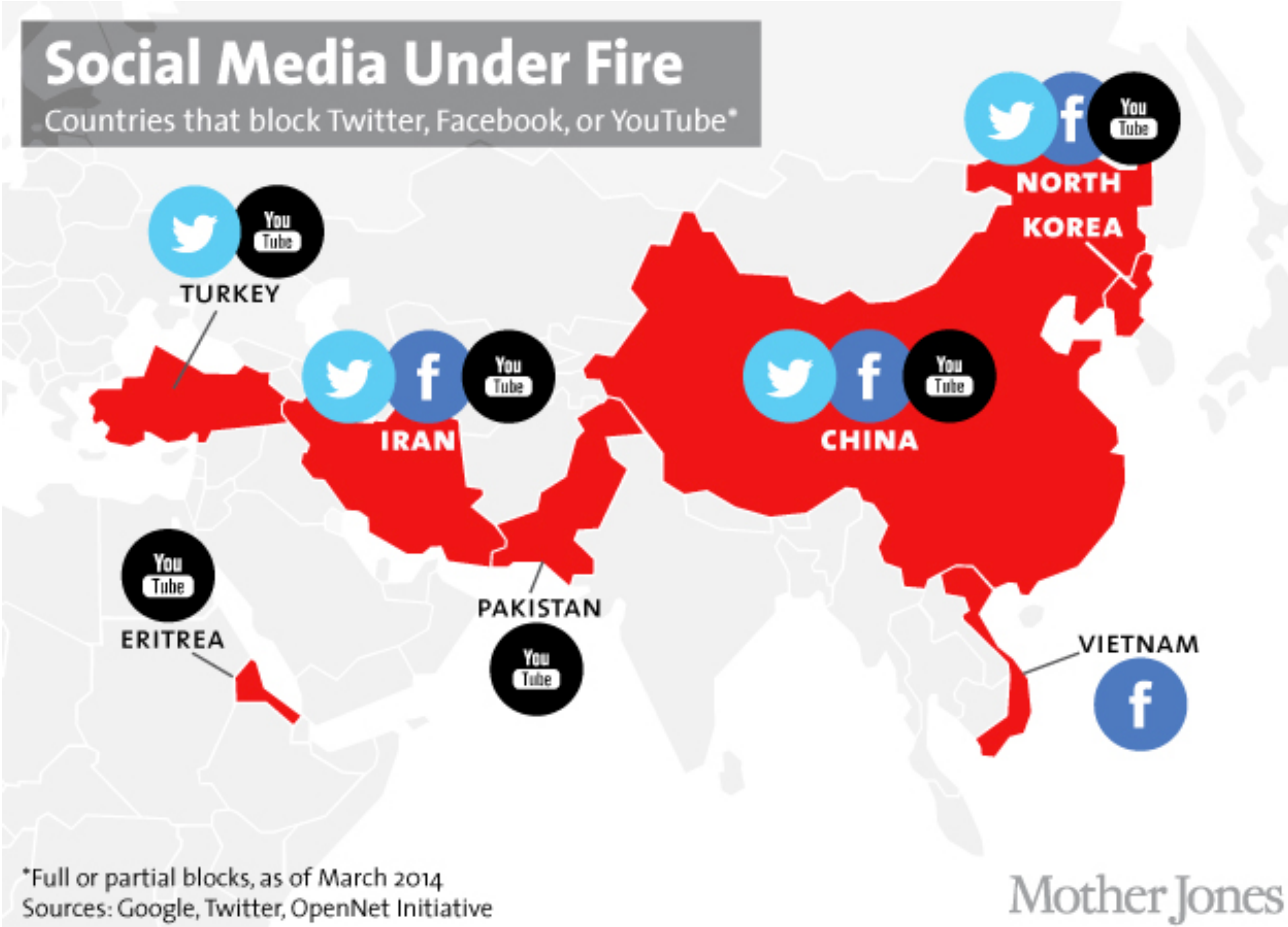
LangID: why needed?

- 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.

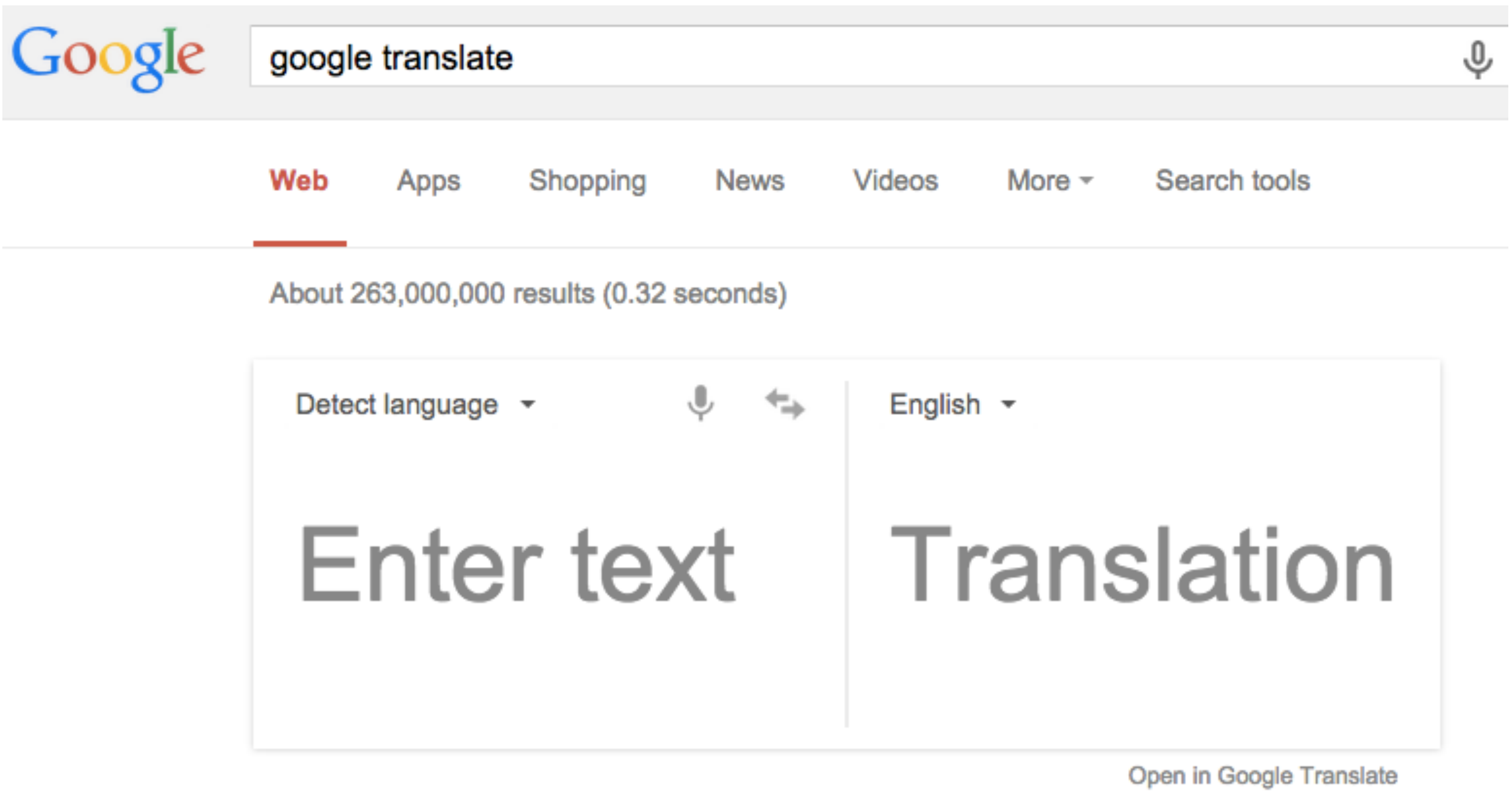




新浪微博
weibo.com

known as the "Chinese Twitter"
120 Million Posts / Day

LangID: Google Translate

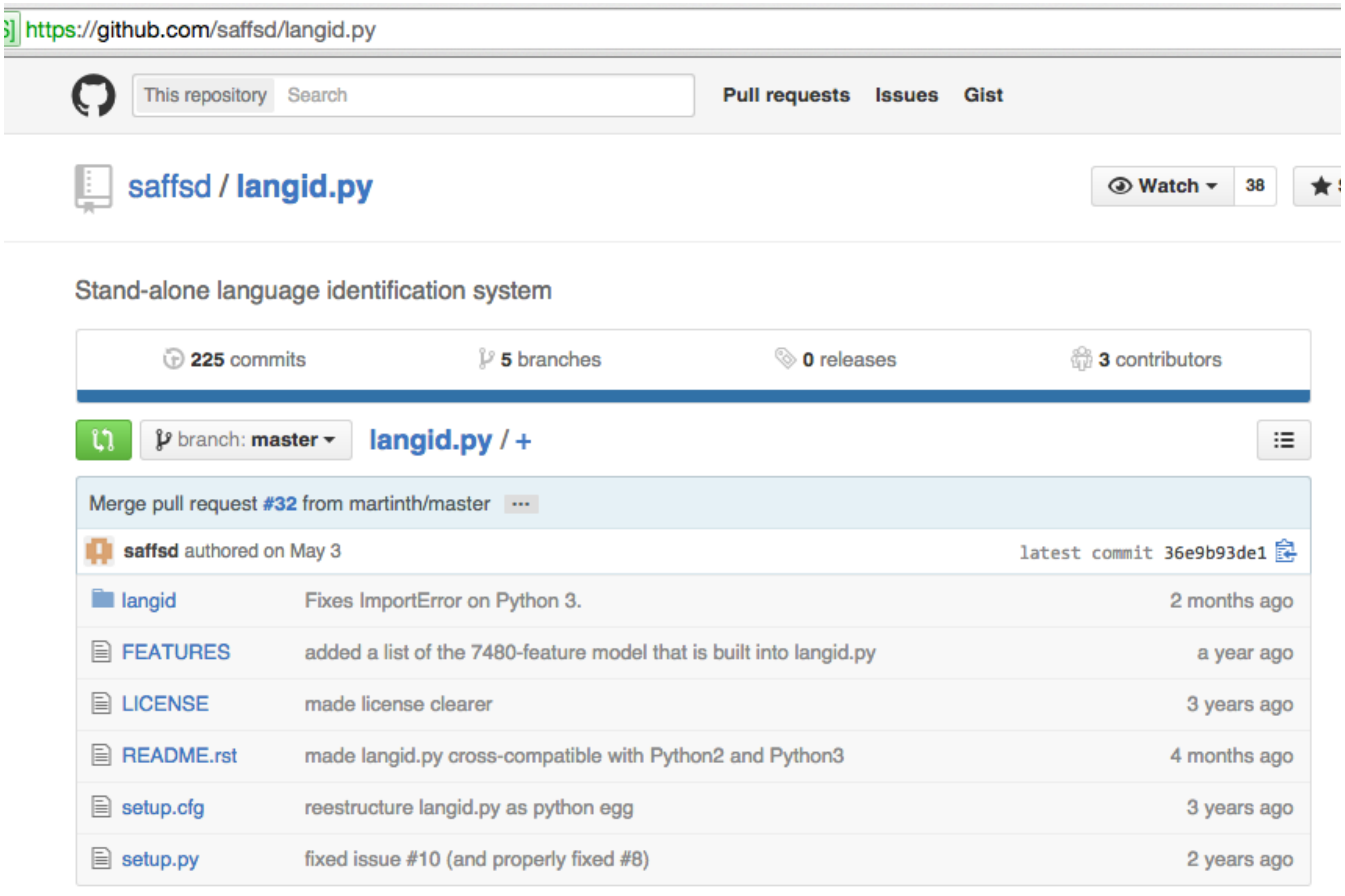


LangID: Twitter API

- 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
  - 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

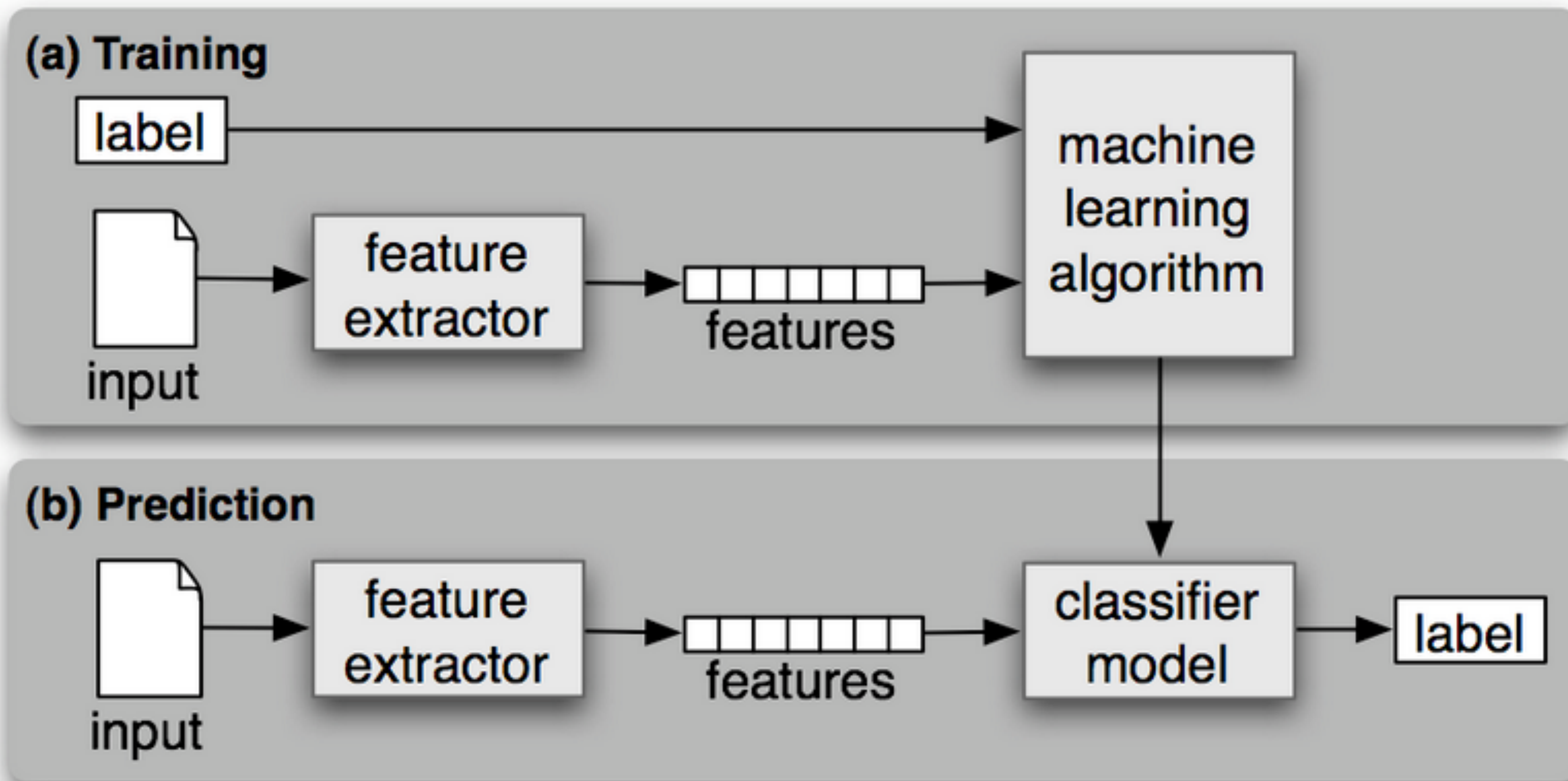
Classification Method:

Supervised Machine Learning

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

Classification Method:

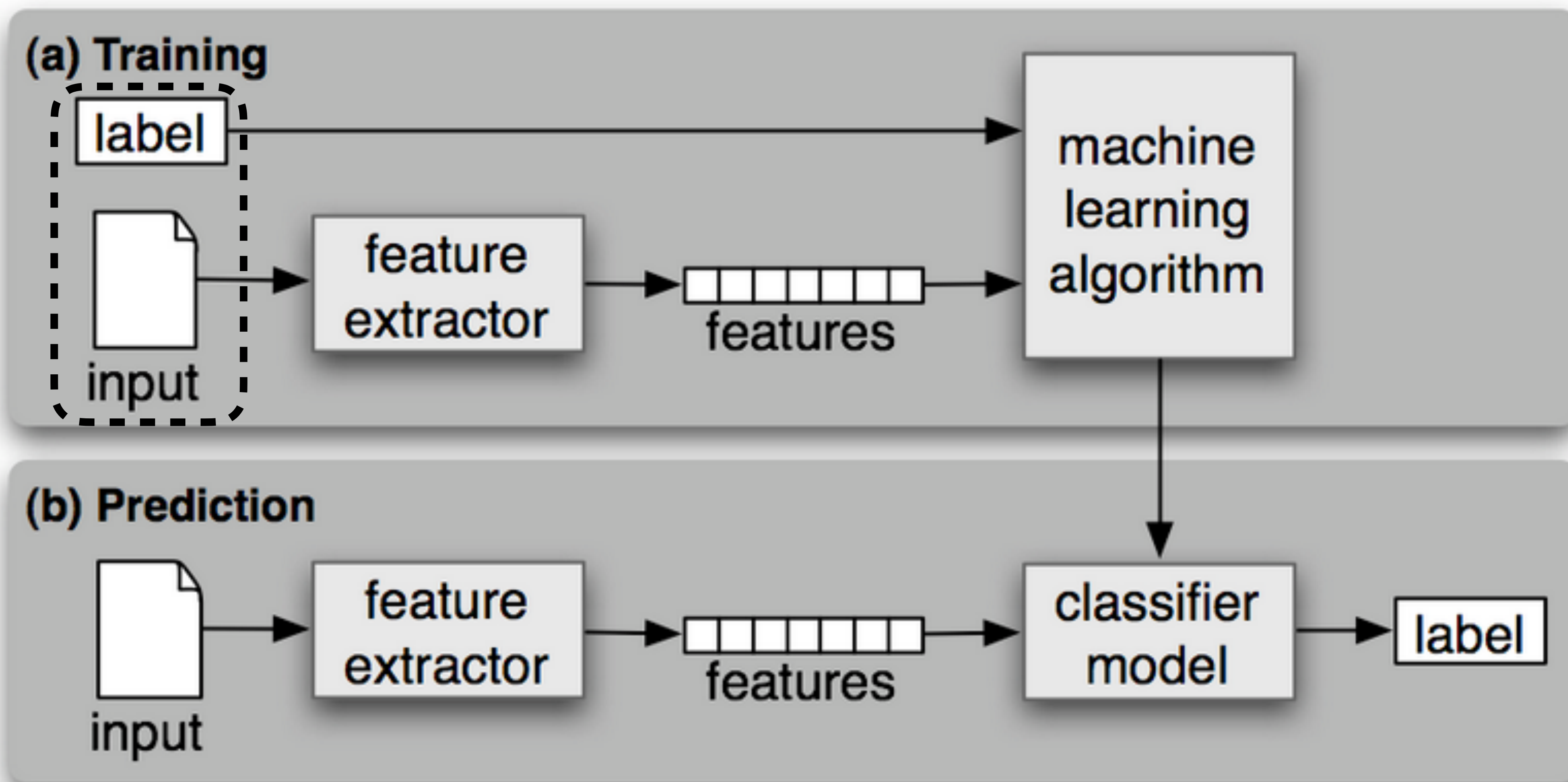
Supervised Machine Learning



Source: NLTK Book

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

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

- Bayes' Theorem:

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

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***:

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

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

← **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 ***t_1, t_2, …, 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? — simple count

Source: adapted from Dan jurafsky

Naïve Bayes

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

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

$O(|T|^n \cdot |C|)$ 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***:

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

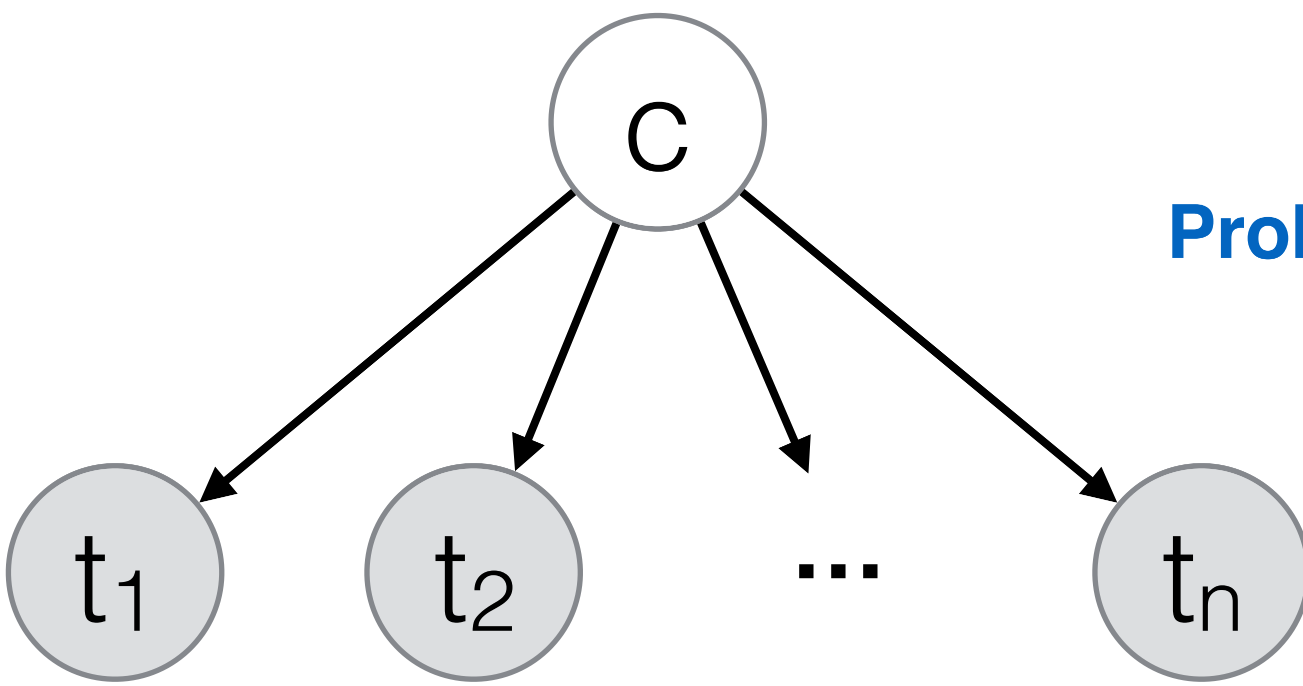
$$\downarrow$$

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

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 features:
  - *Multinomial: discrete features*
  - *Bernoulli: binary features*
  - *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: binary features*
  - *Gaussian: continuous features*

Source: adapted from Dan jurafsky

LangID features

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

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

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

- 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

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

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

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

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: (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

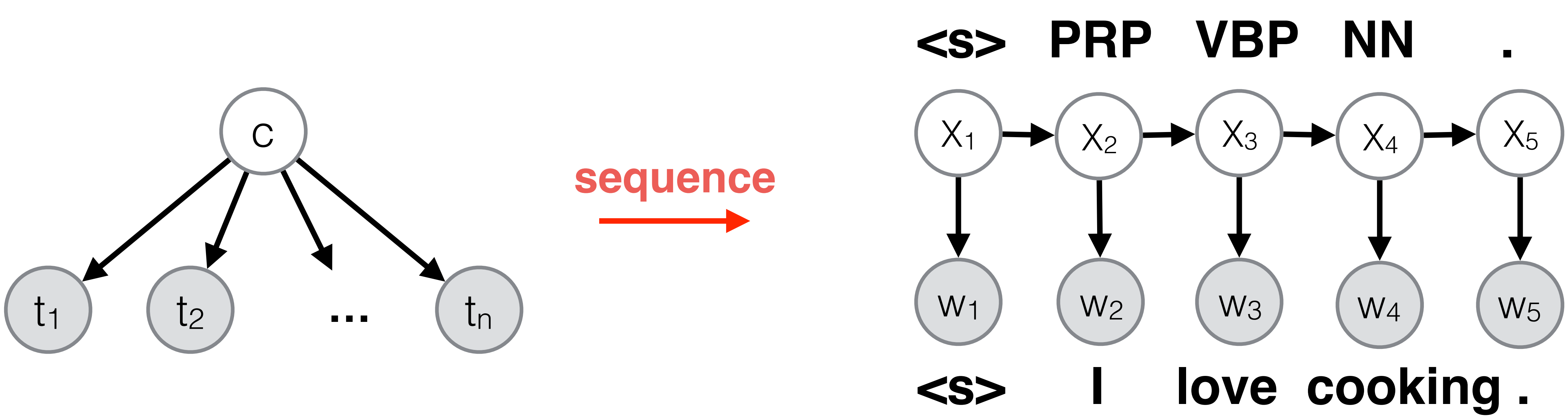
Correlated Features

- 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:
  - **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

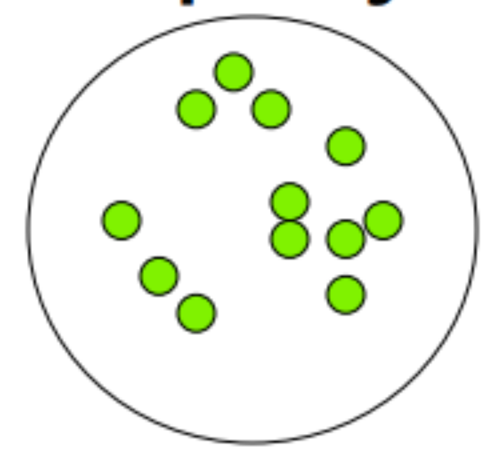
Source: Lui and Baldwin "langid.py: An Off-the-shelf Language Identification Tool" ACL 2012

Entropy & Information Gain

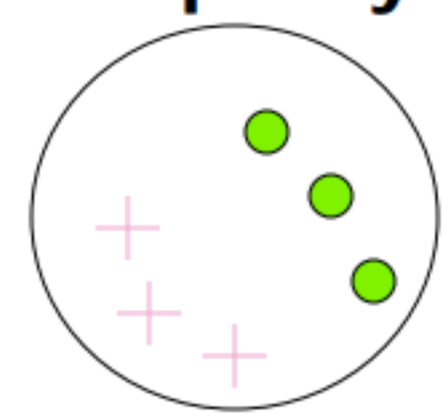
- **Entropy** is a measure of disorder in a dataset

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

H(X) = 0
Minimum impurity



H(X) = 1
Maximum impurity

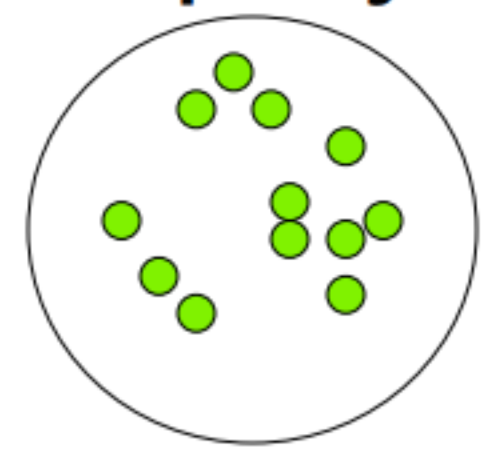


Entropy & Information Gain

- **Entropy** is a measure of disorder in a dataset

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

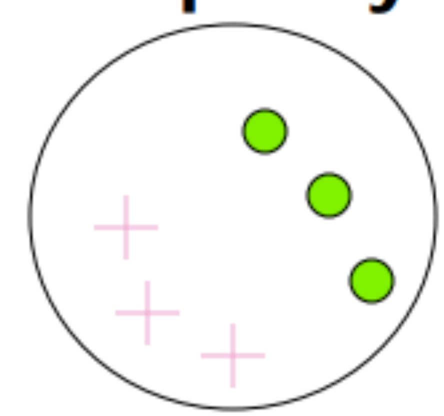
H(X) = 0
Minimum impurity



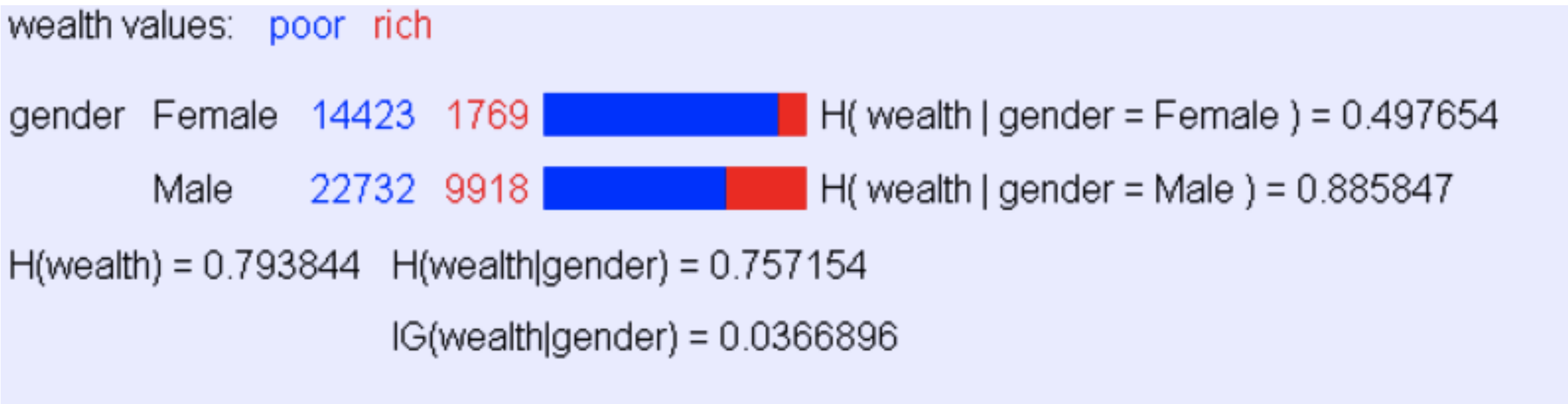
- **Information Gain** is a measure of the decrease in disorder achieved by partitioning the original data set.

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

H(X) = 1
Maximum impurity



Information Gain



$$H(X) = -\sum_i P(x_i) \log P(x_i) \qquad IG(Y \mid X) = H(Y) - H(Y \mid X)$$

Source: Andrew Moore

Information Gain

wealth values: poor rich

		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

Information Gain used for?

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

Wealth

IG(wealth|gender) = 0.0366896

IG(wealth|agegroup) = 0.0843813

Longevity

IG(LongLife | HairColor) = 0.01

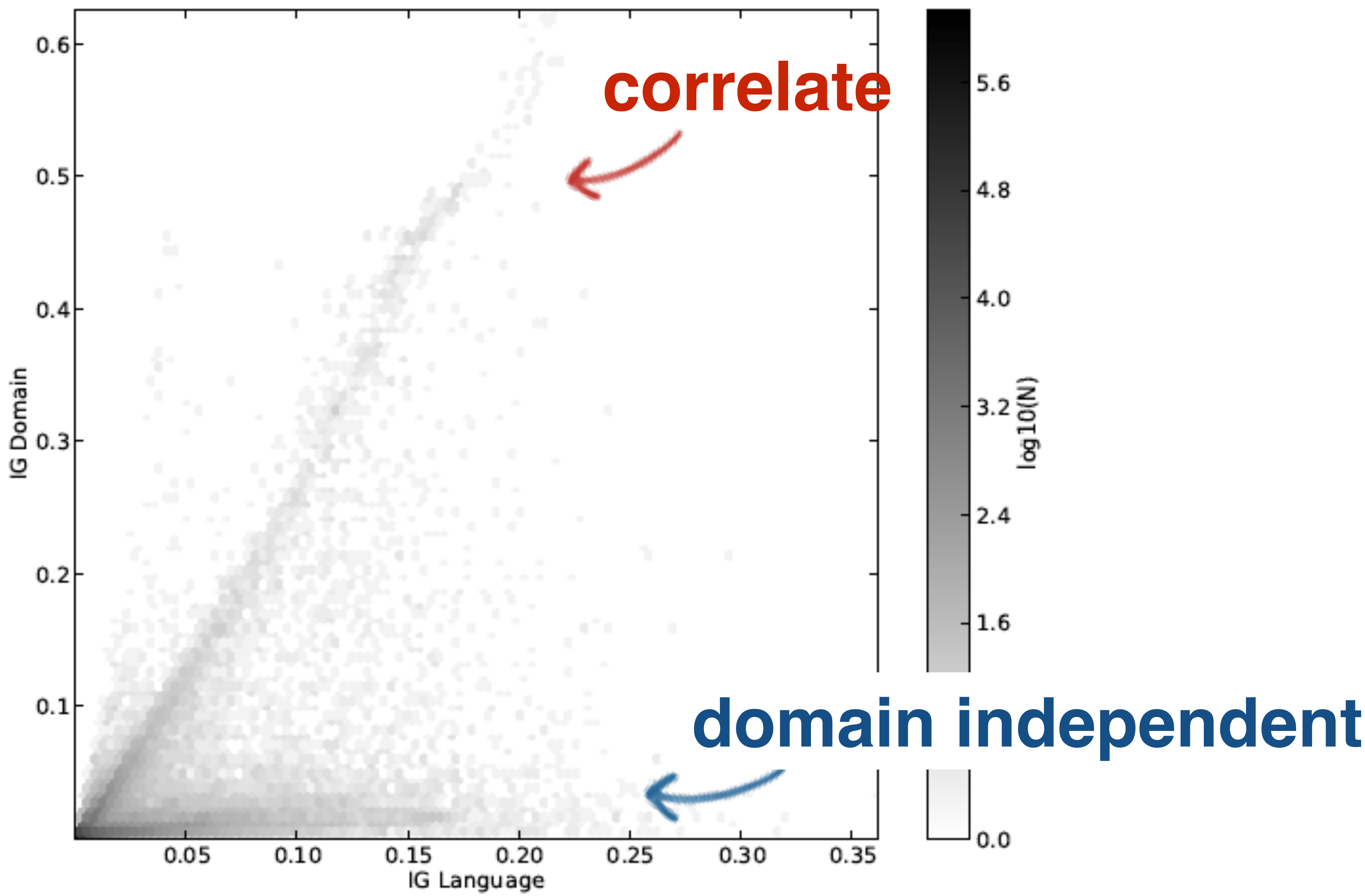
IG(LongLife | Smoker) = 0.2

IG(LongLife | Gender) = 0.25

IG(LongLife | LastDigitOfSSN) = 0.00001

LangID Tool: langid.py

- feature selection using Information Gain (IG)



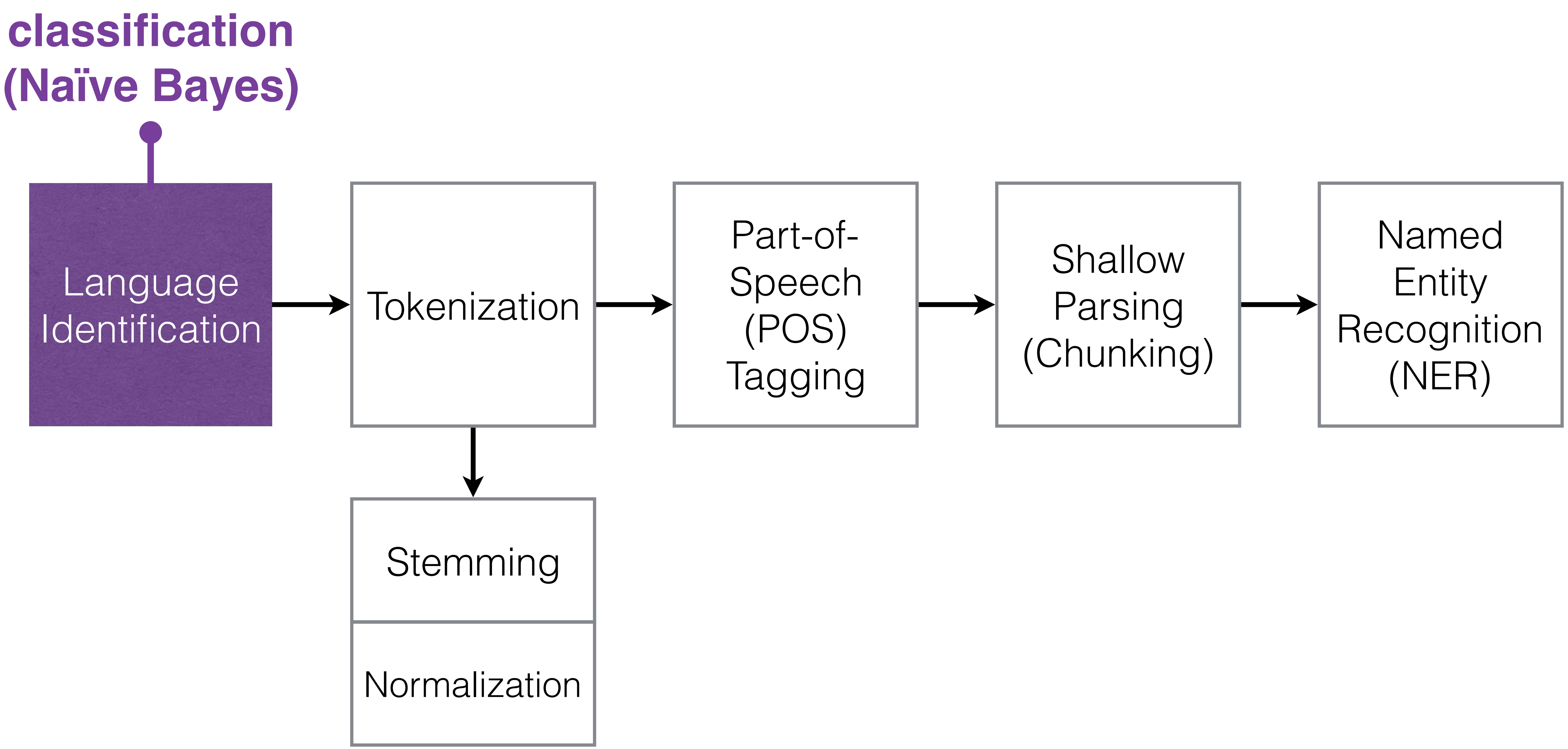
Source: Lui and Baldwin "langid.py: An Off-the-shelf Language Identification Tool" ACL 2012

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

Summary


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

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

socialmedia-class.org