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

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?

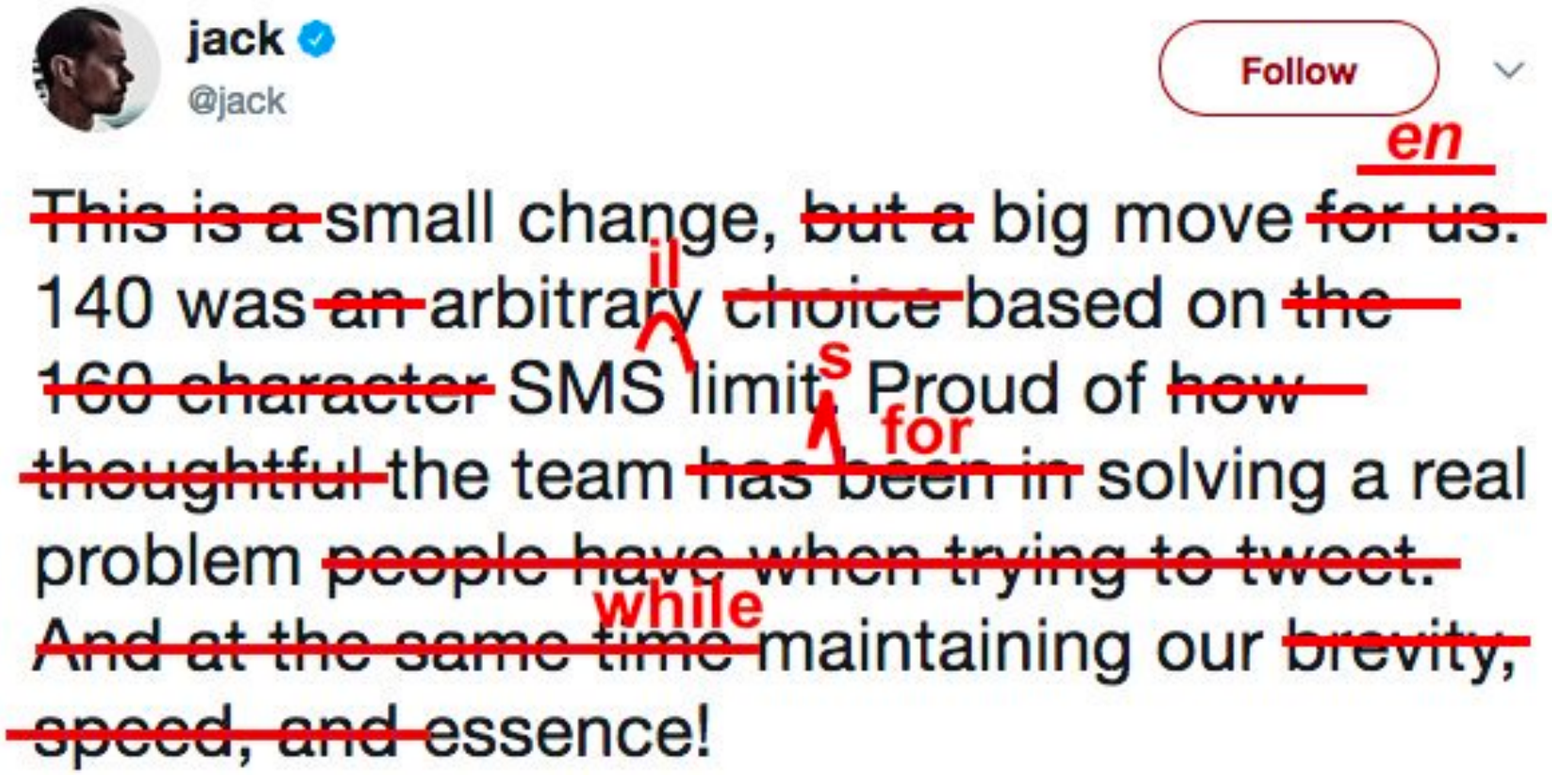
News
Blogs
Wikipedia
Forums
Comments
Twitter
…

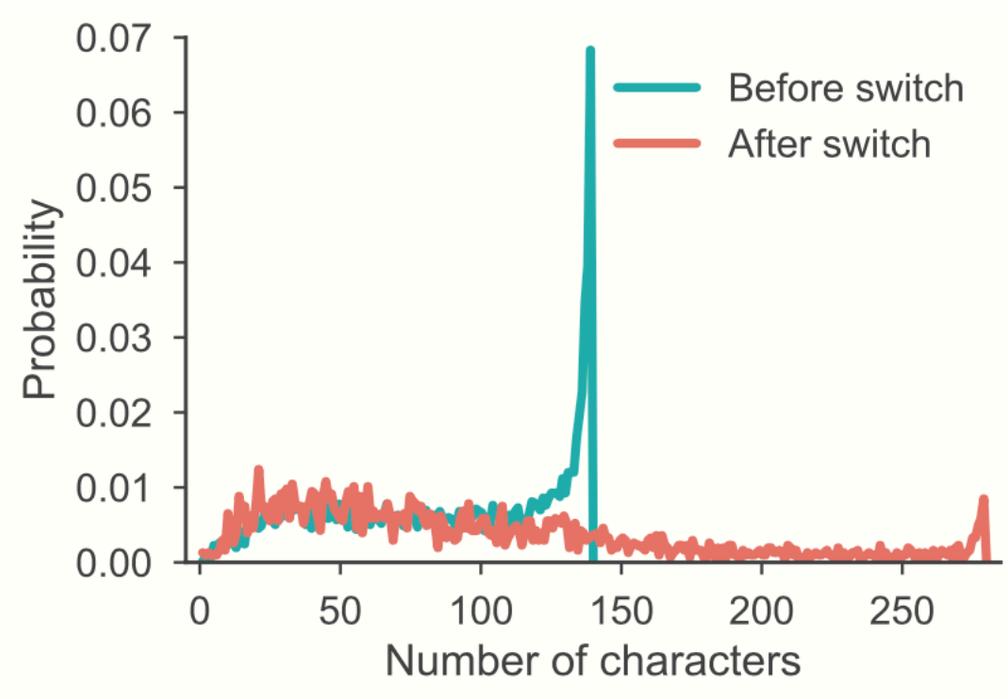
Domain/Genre

- 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





Source: Gligoric et al.
"How Constraints Affect Content: The Case of Twitter's Switch from 140 to 280 Characters" ICWSM 2018

Domain/Genre

- How different?

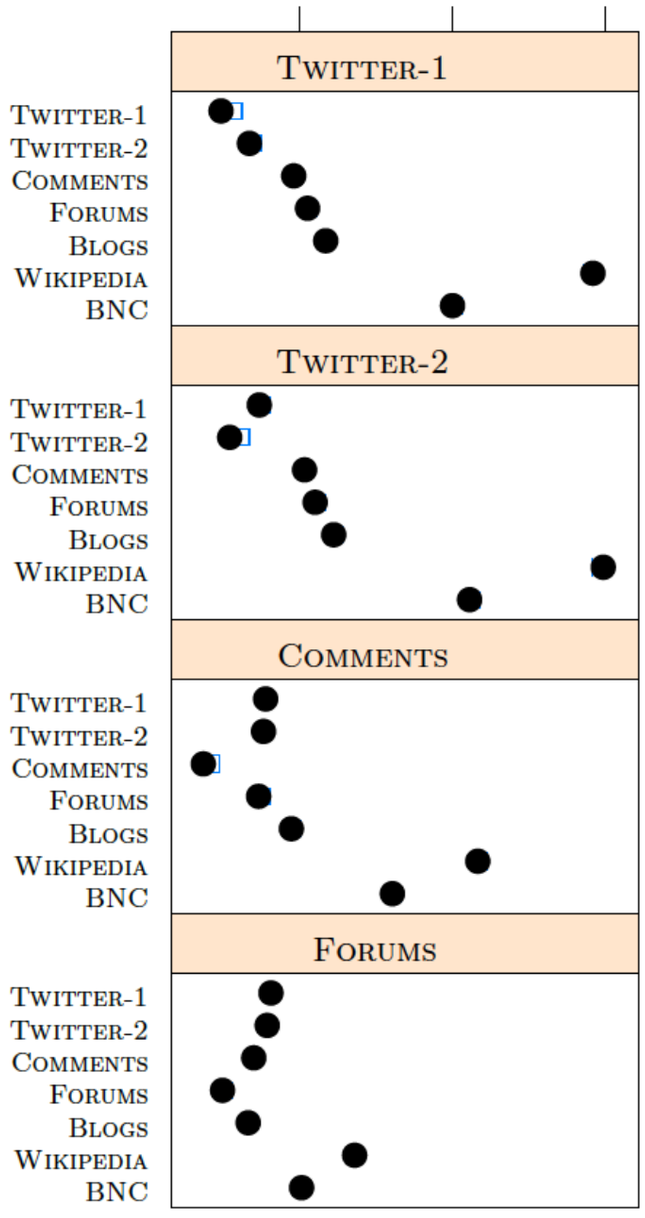
out-of-vocabulary

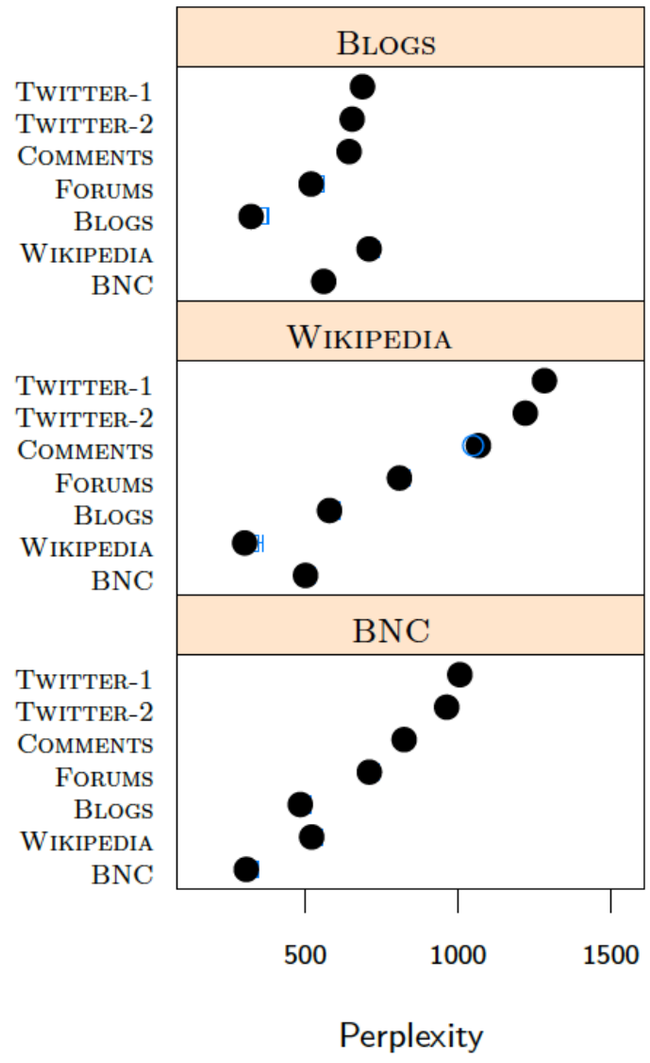
Corpus	Word length	Sentence length	%OOV
TWITTER-1	3.8±2.4	9.2±6.4	**24.6**
TWITTER-2	3.8±2.4	9.0±6.3	**24.0**
COMMENTS	3.9±3.2	10.5±10.1	**19.8**
FORUMS	3.8±2.3	14.2±12.7	**18.1**
BLOGS	4.1±2.8	18.5±24.8	**20.6**
WIKIPEDIA	4.5±2.8	21.9±16.2	**19.0**
BNC	4.3±2.8	19.8±14.5	**16.9**

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)

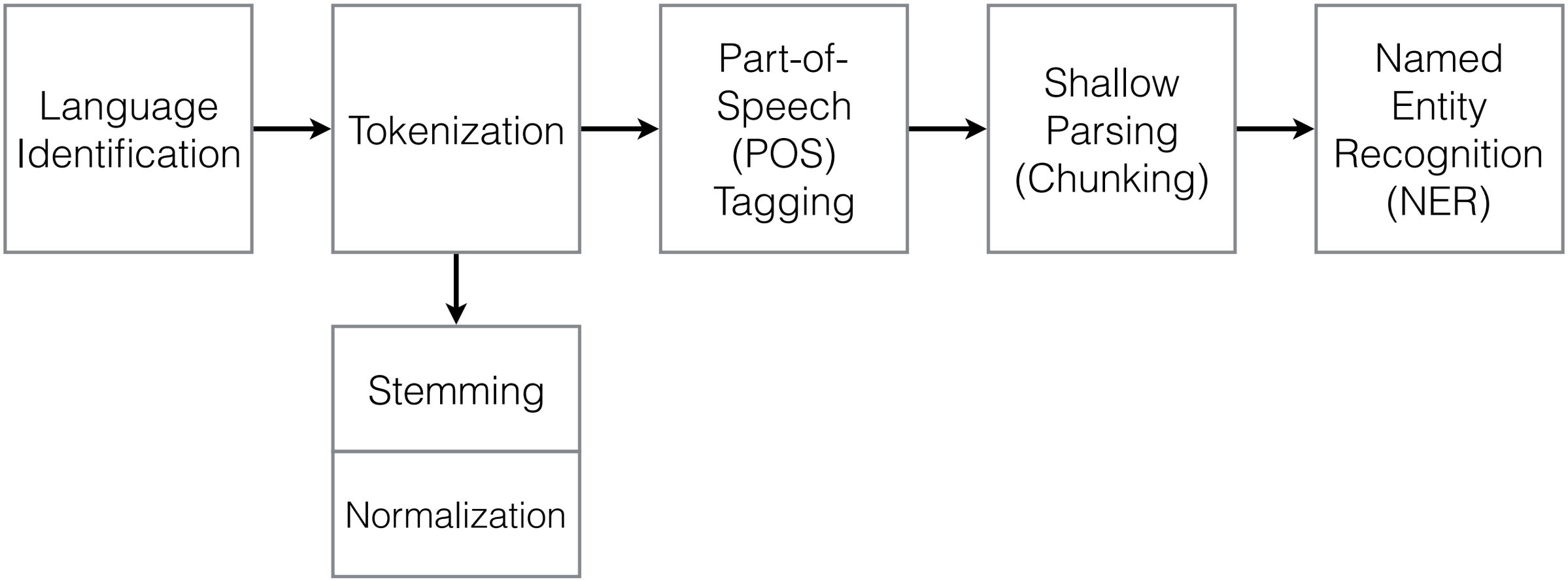
  - large amount of data

  - large demand for research/commercial purpose

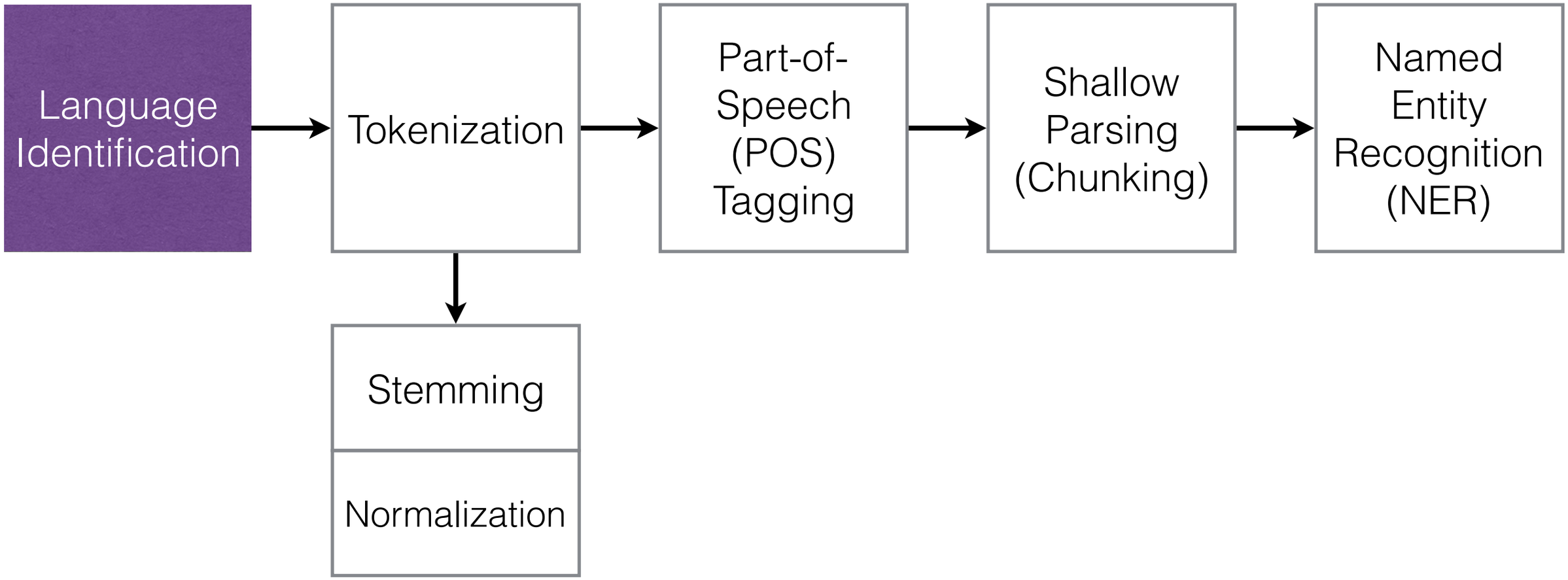
  - too different from well-edited text (which most NLP tools have been made for)

NLP Pipeline

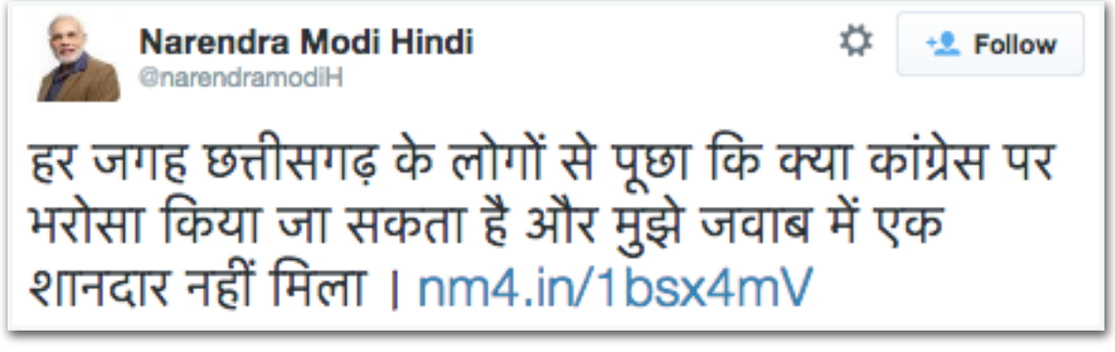
NLP Pipeline

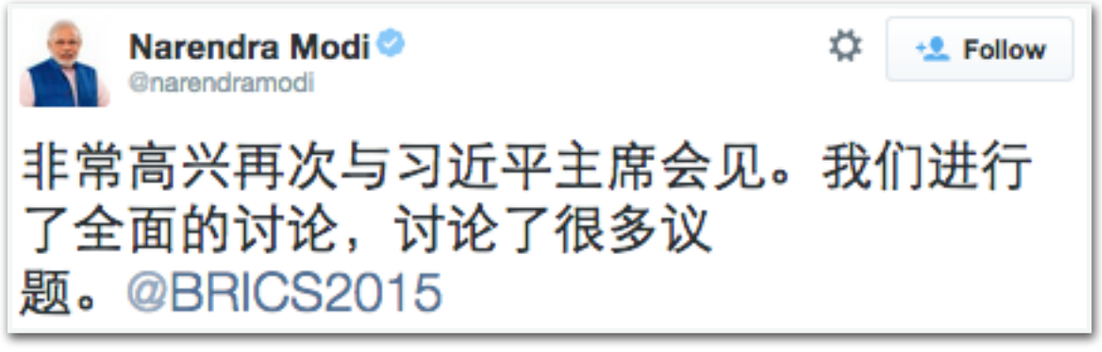


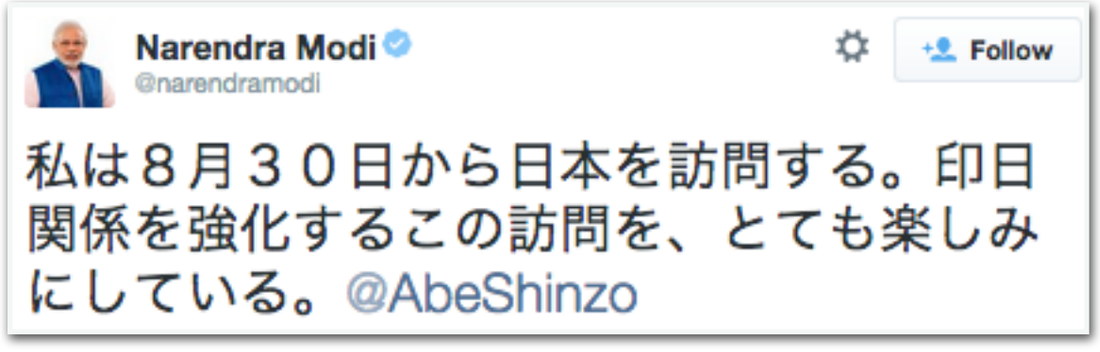
NLP Pipeline



Language Identification
(a.k.a Language Detection)





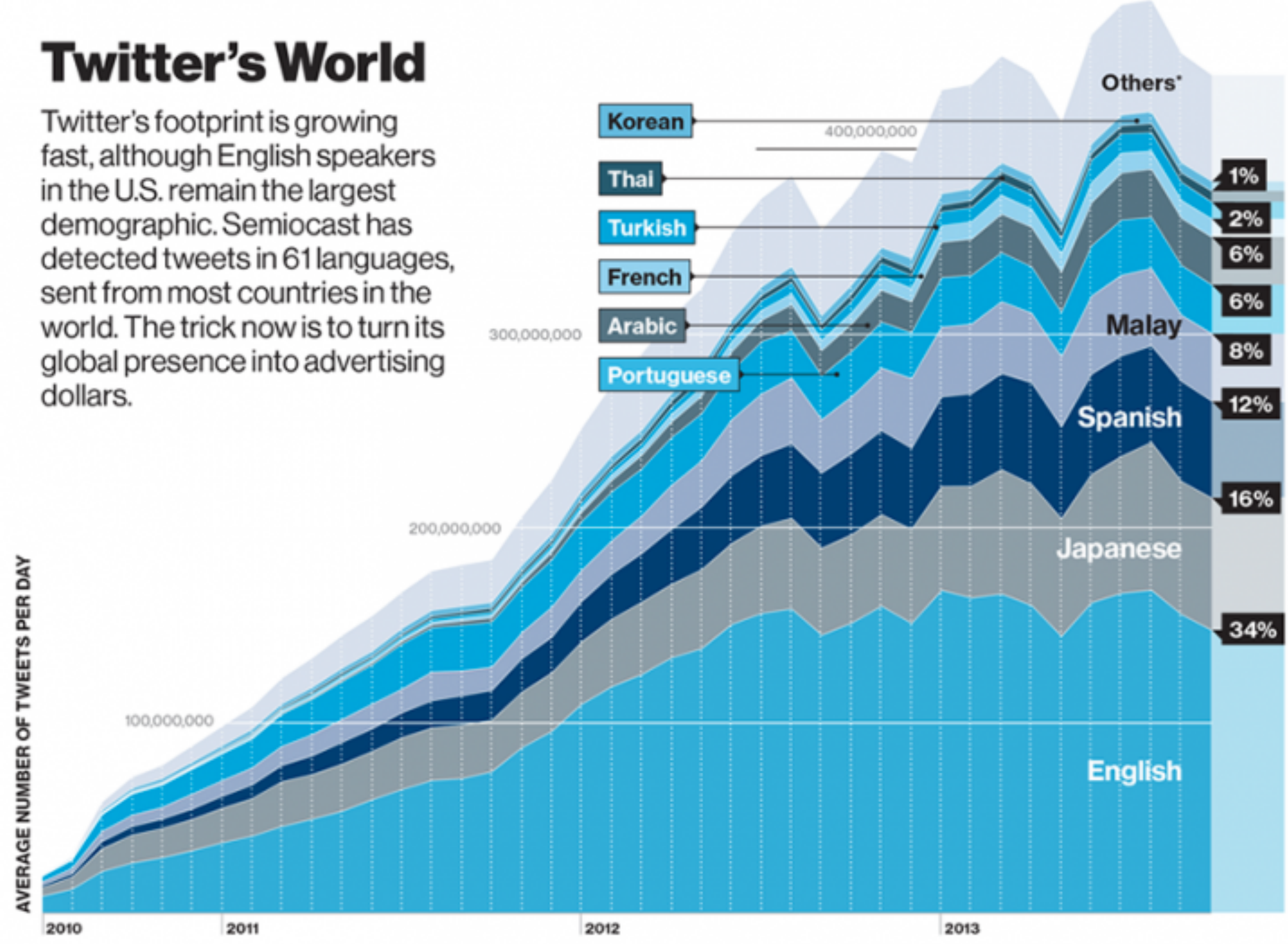


LangID: why needed?

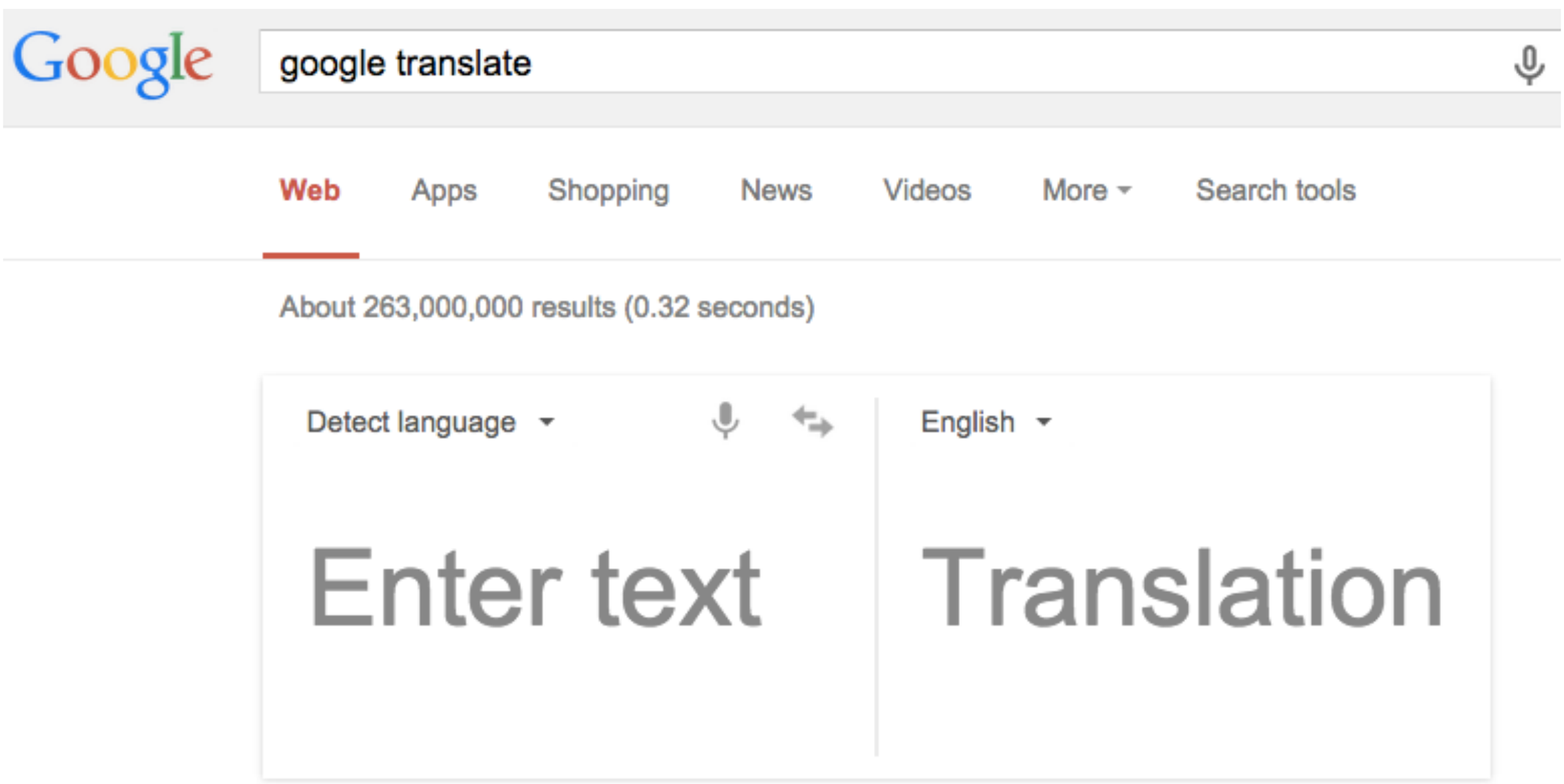
- Twitter is highly multilingual
- But NLP is often monolingual

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.



LangID: Google Translate



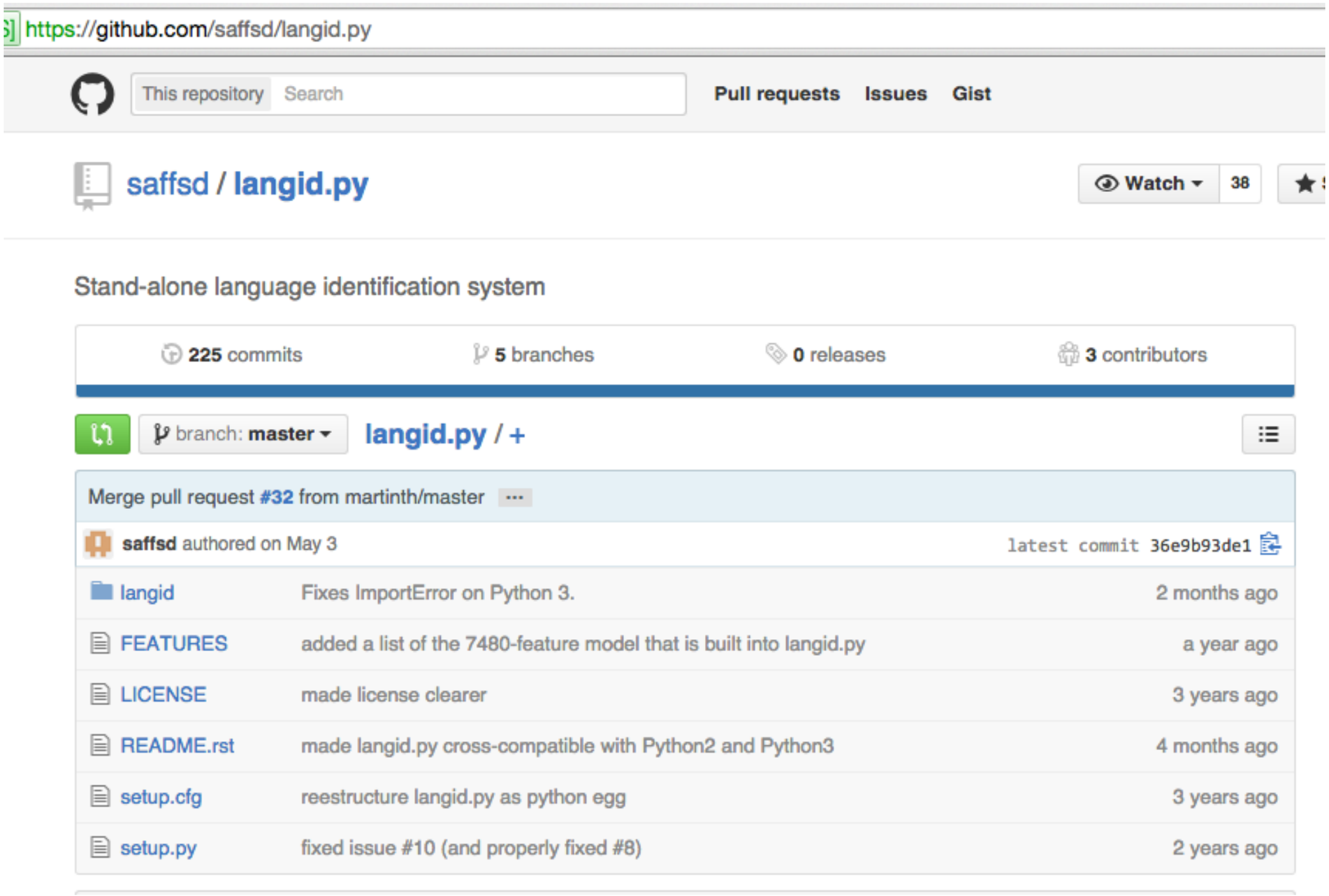
LangID: Twitter API

- 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', 0.5713348767990067 4)
>>> langid.set_languages(['de','fr','it'])
>>> langid.classify("I do not speak english")
('it', 0.9999983579147845 3)
>>> langid.set_languages(['en','it'])
>>> langid.classify("I do not speak english")
('en', 0.9917619037875037 3)
```

LangID:

A Classification Problem

- Input:
  - a document ***d***
  - a fixed set of classes ***C = {c1, c2, …, cj}***

- Output:
  - a predicted class ***c*** ∈ ***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:

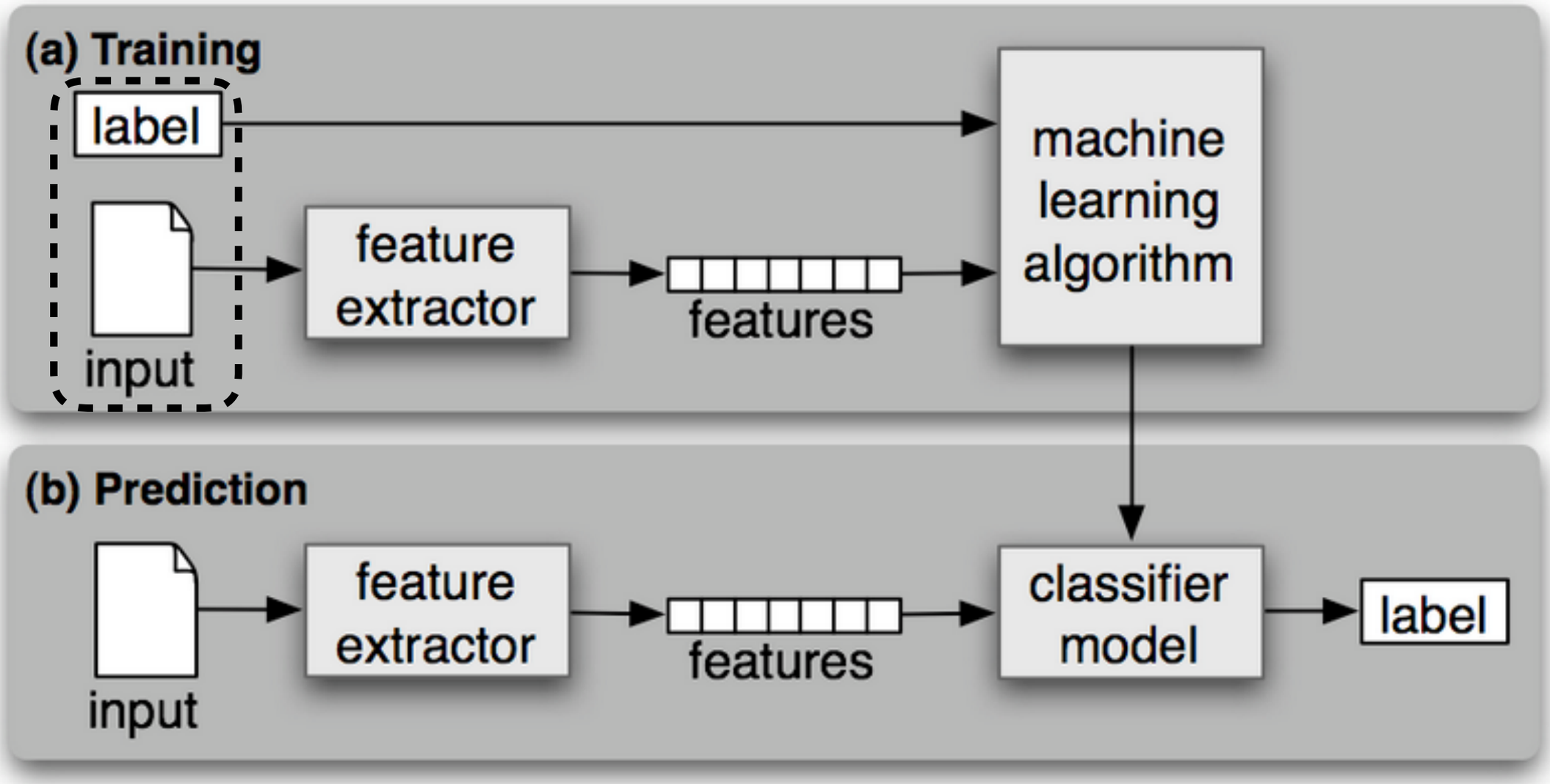
Supervised Machine Learning

- Input:
  - a document ***d***
  - a fixed set of classes ***C = {c1, c2, …, cj}***
  - a training set of ***m*** hand-labeled documents ***(d1, c1), … , (dm, cm)***

- Output:
  - a learned classifier ***γ: d → 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

- 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***:

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

maximum a posteriori

Source: adapted from Dan jurafsky

Naïve Bayes

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

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

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

← **Bayes Rule**

Source: adapted from Dan jurafsky

Naïve Bayes

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

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

$$= \arg\max_{c \in C} \frac{P(d \mid c)P(c)}{P(d)} \quad \leftarrow \textbf{Bayes Rule}$$

$$= \arg\max_{c \in C} P(d \mid c)P(c) \quad \leftarrow \textbf{drop the denominator}$$

Source: adapted from Dan jurafsky

Naïve Bayes

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

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

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

Source: adapted from Dan jurafsky

Naïve Bayes

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

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

prior

how often does this class occur? — simpe count

Source: adapted from Dan jurafsky

Naïve Bayes

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

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

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

$$P(t_1, t_2, ..., t_n \mid c)$$

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

Source: adapted from Dan jurafsky

Naïve Bayes

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

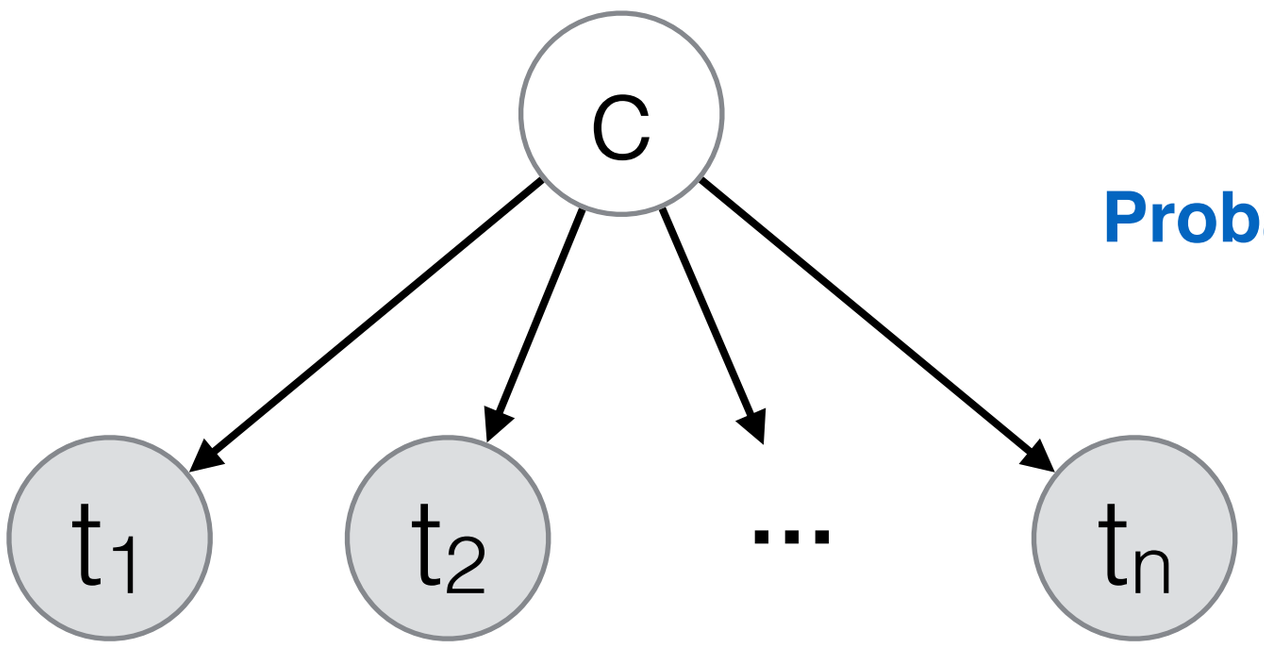
$$c_{MAP} = \arg\max_{c \in C} P(t_1, t_2, ..., t_n \mid c) P(c)$$

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

Source: adapted from Dan jurafsky

Naïve Bayes

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



Probabilistic Graphical Model

Variations of Naïve Bayes

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

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

Source: adapted from Dan jurafsky

Variations of Naïve Bayes

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

- 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

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

- 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	?

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

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

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

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-weight
  - often works well on small datasets
  - 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)**

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', 0.5713348767990067 4)
>>> langid.set_languages(['de','fr','it'])
>>> langid.classify("I do not speak english")
('it', 0.9999983579147845 3)
>>> langid.set_languages(['en','it'])
>>> langid.classify("I do not speak english")
('en', 0.9917619037875037 3)
```

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

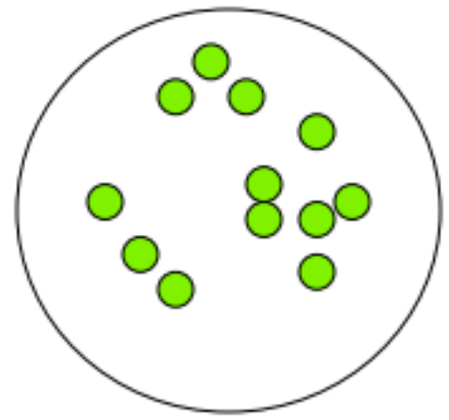
Source: Lui and Baldwin "langid.py: An Off-the-shelf Language Identification Tool" ACL 2012

Entropy & Information Gain

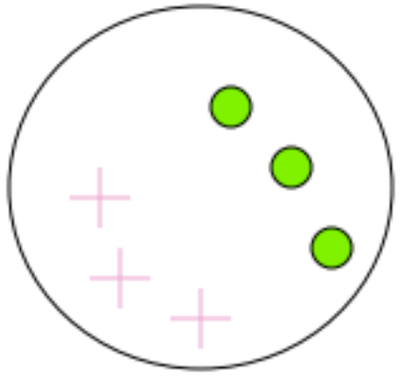
- **Entropy** is a measure of disorder in a dataset

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

H(X) = 0
Minimum impurity



H(X) = 1
Maximum impurity

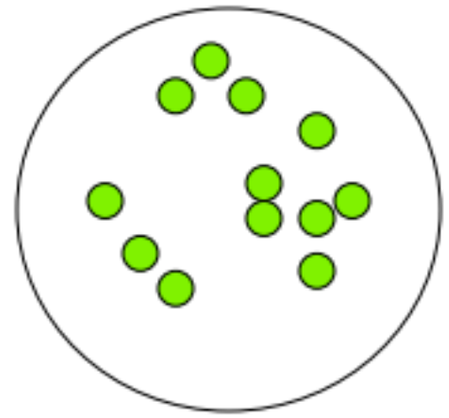


Entropy & Information Gain

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

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

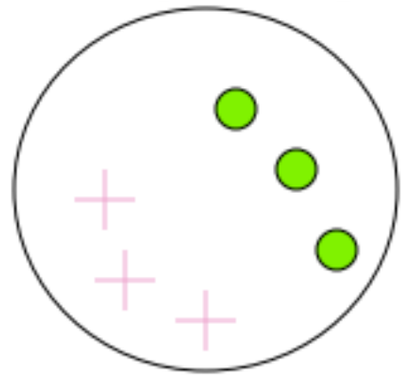
H(X) = 0
Minimum impurity



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

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

H(X) = 1
Maximum impurity



Information Gain

wealth values: poor rich

gender Female 14423 1769 H(wealth | gender = Female) = 0.497654

Male 22732 9918 H(wealth | gender = Male) = 0.885847

H(wealth) = 0.793844 H(wealth|gender) = 0.757154

IG(wealth|gender) = 0.0366896

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

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

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

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

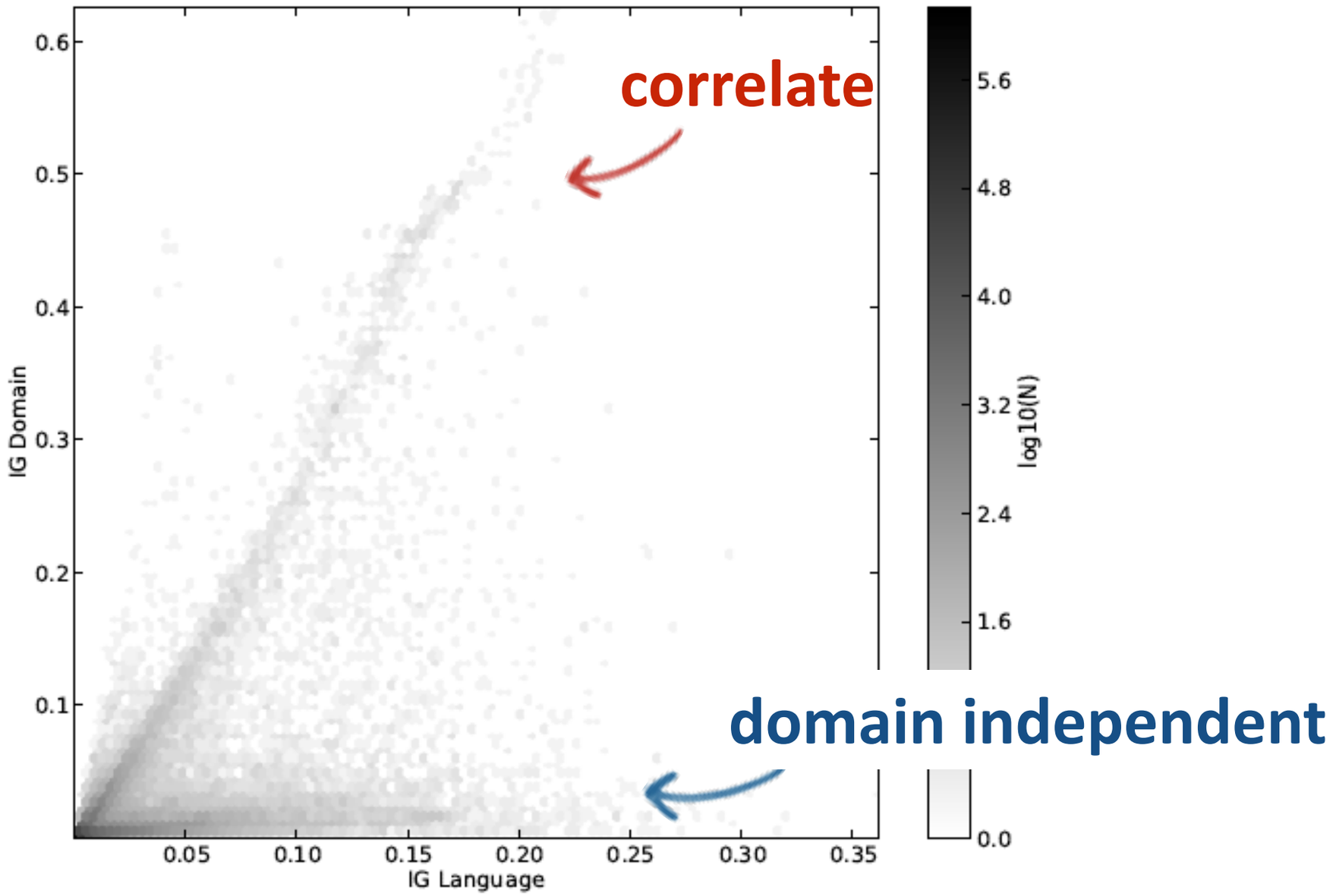
IG(LongLife | Smoker) = 0.2

IG(LongLife | Gender) = 0.25

IG(LongLife | LastDigitOfSSN) = 0.00001

LangID Tool: langid.py

- feature selection using Information Gain (IG)



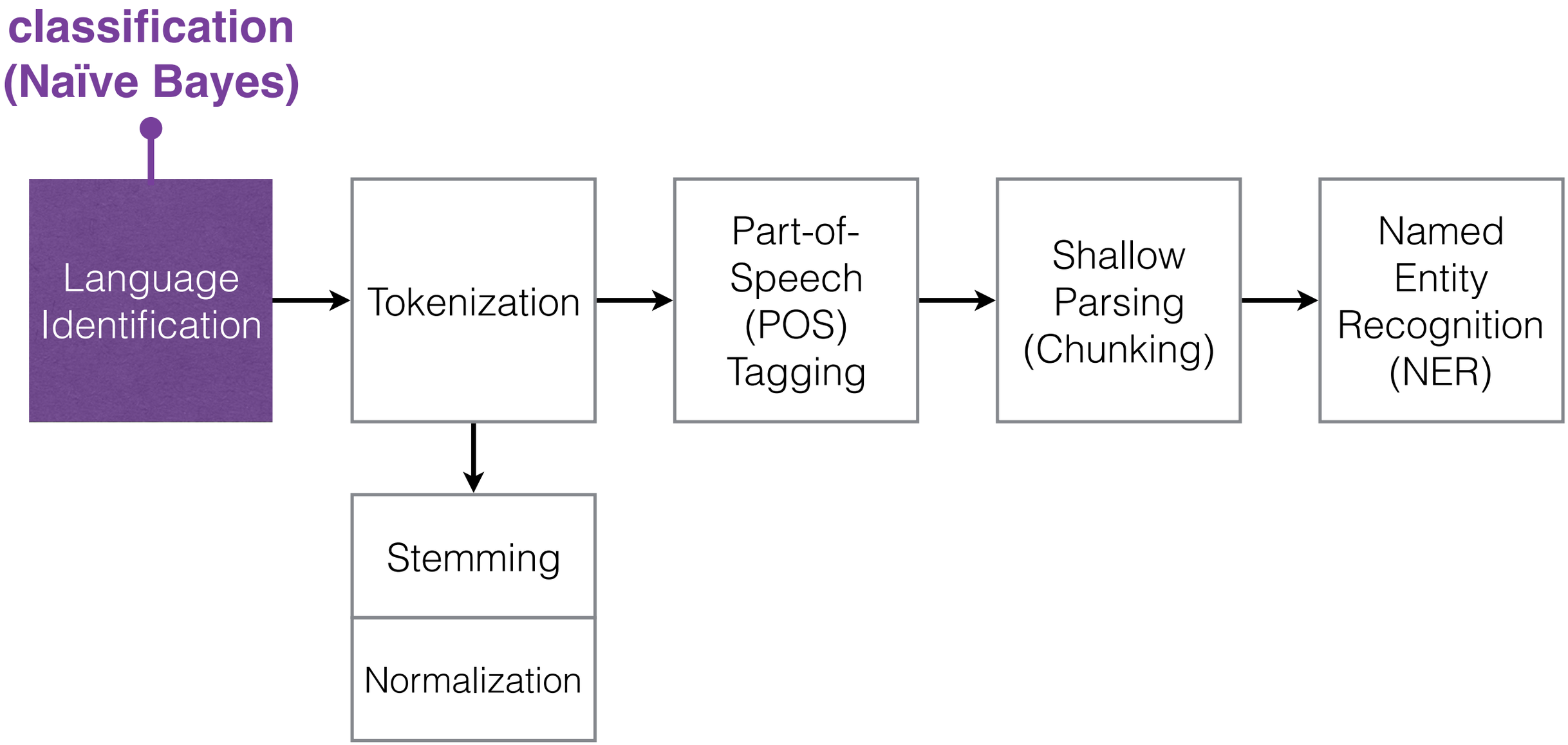
Source: Lui and Baldwin "langid.py: An Off-the-shelf Language Identification Tool" ACL 2012

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
Stemming
Normalization
Part-of-Speech (POS) Tagging
Shallow Parsing (Chunking)
Named Entity Recognition (NER)