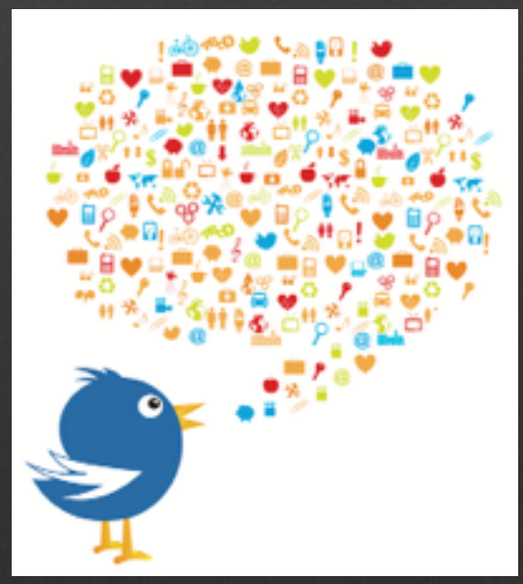
Automatic Language Classification

Authors: Daniel Horowitz, Jeroni Carandell, Iosu Mendizabal

Motivation

- Before starting any NLP application you need to know…

…what language you are going to target.



Our Approach

- Database of tweets
- Classification via N-gram models
  - Lidstone smoothing
  - Ranking models

Database - Parse

The Cloud Application Platform

- Filled using:
- Real twitter posts:
    - 4 languages: EN, ES, FR, PT.
- Tagged them using the API of google translate.

- Parse. com
  - Authentication
  - App
- Twitter API
  - Authentication
  - Search
- Refining the results

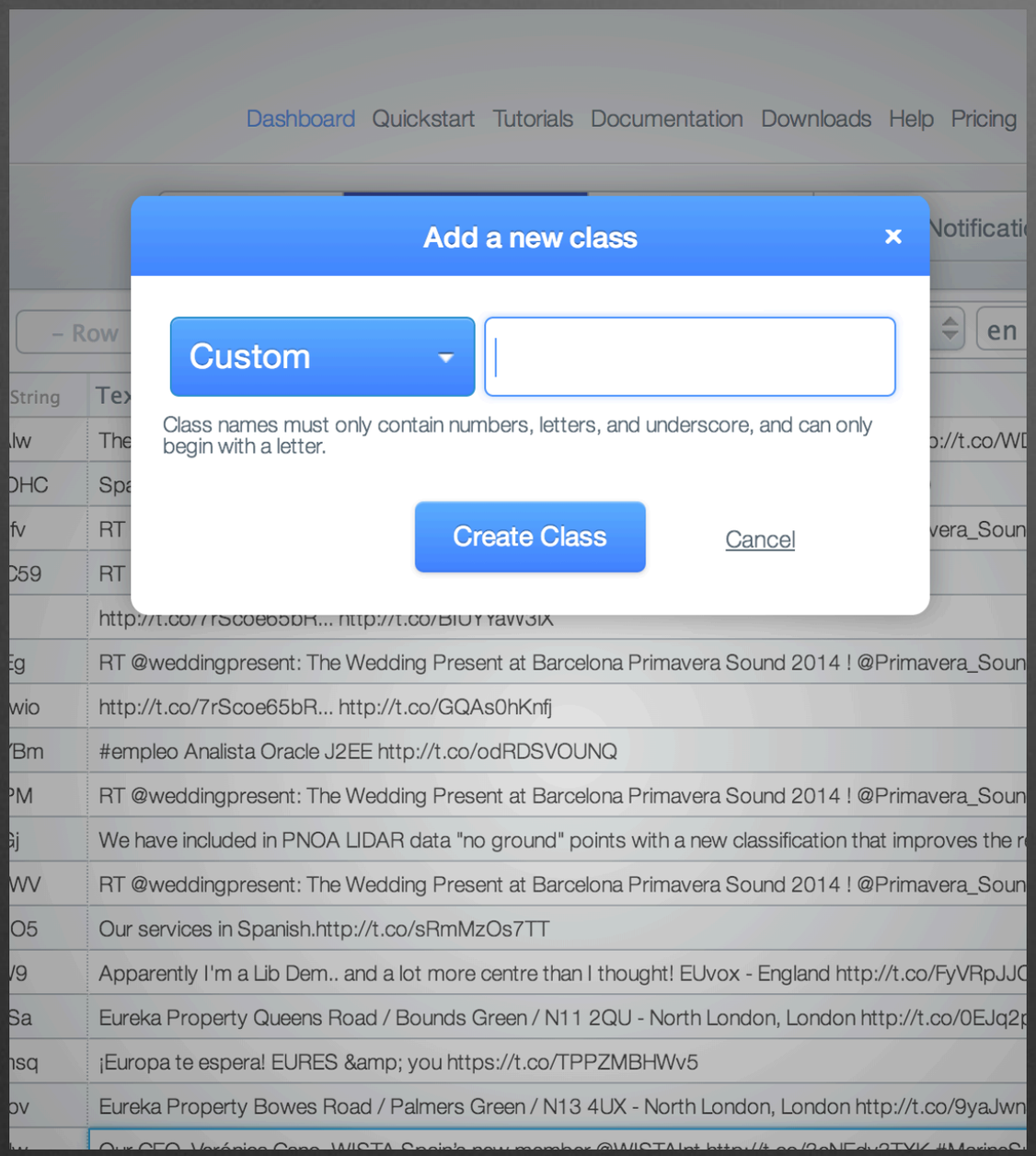
Parse

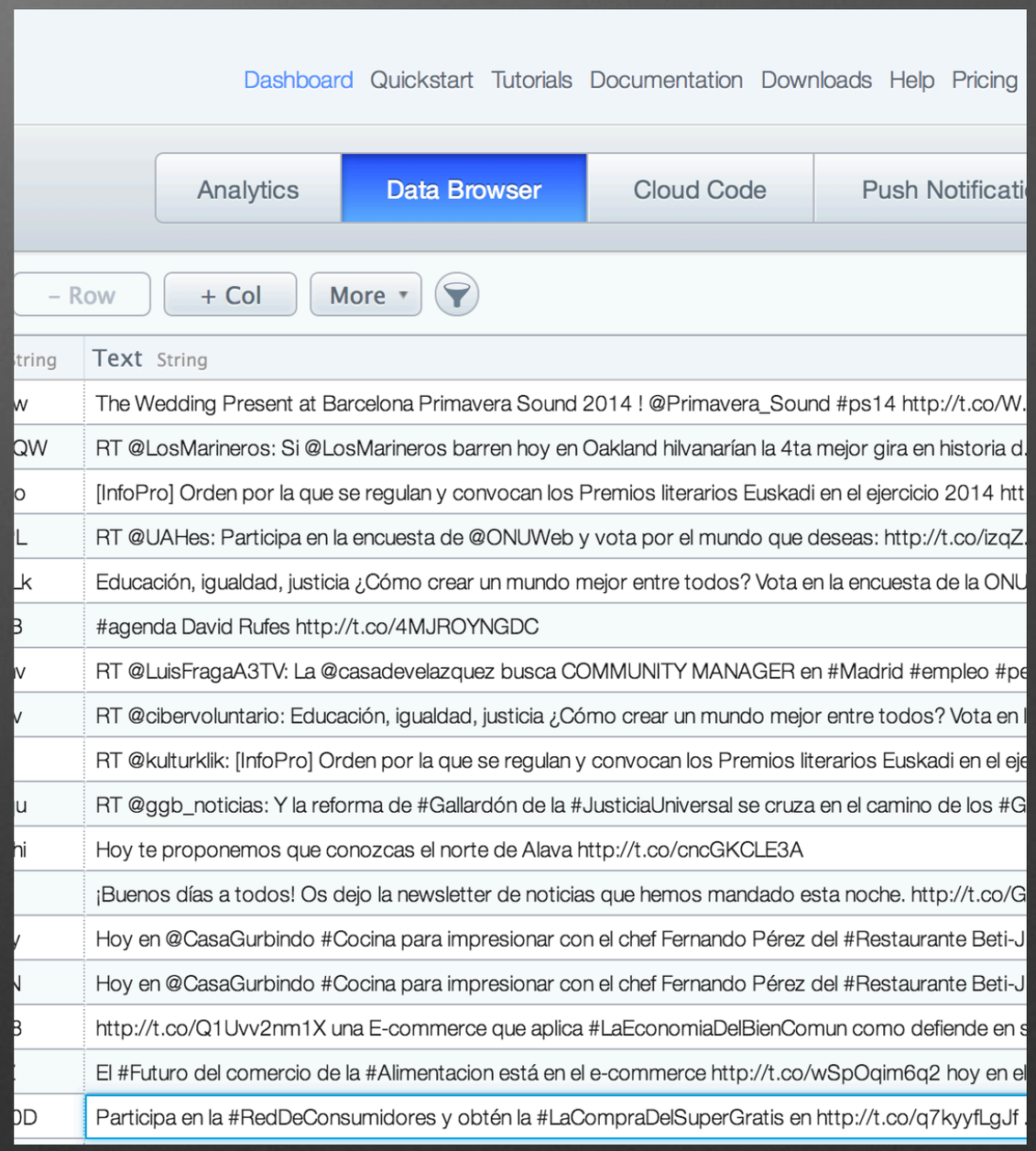
The Cloud Application Platform

- Motivation
  - Easy to use
  - Web based (Multiplatform)
    - Well integrated, with most of the programming languages
- Requirements
  - Accounts
  - App (API credentials)

Parse

The Cloud Application Platform





Twitter API

- Authentication
  - Register as twitter developer
    - App (API credentials)
    - OAUTH 2 protocol
- Search
  - Language parameter (beta version)

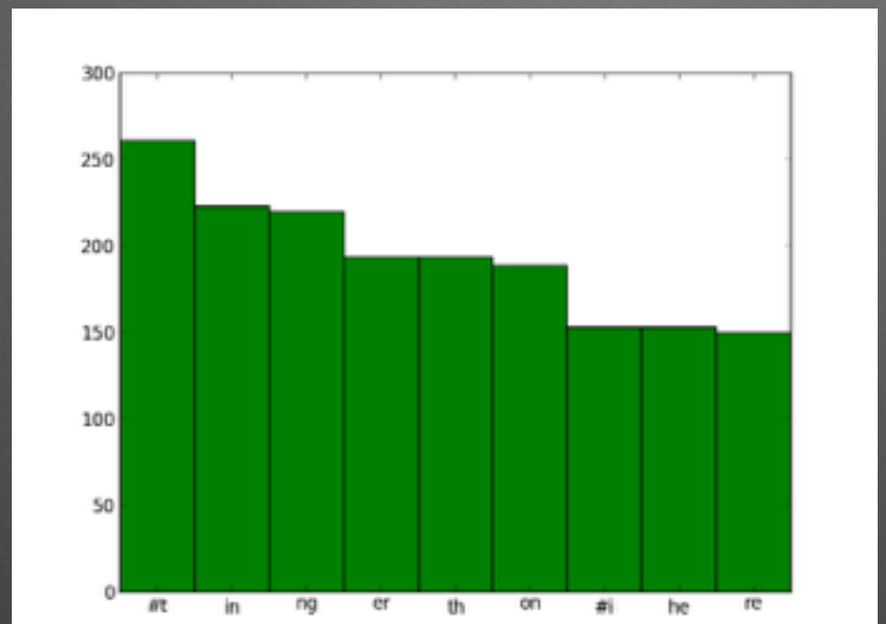
Refining the results

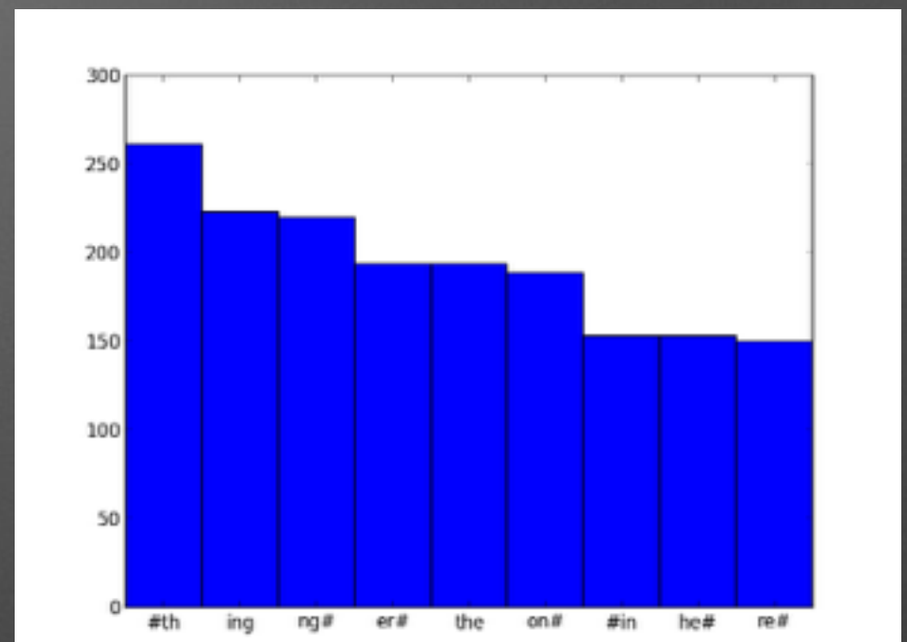
- Google translate API for labeling
- Regular expressions
  - URL’s
  - RT tags
  - Duplicated spaces
  - Accents, symbols and especial characters
- Duplicated tweets
  - Re-tweets
  - Advertisement

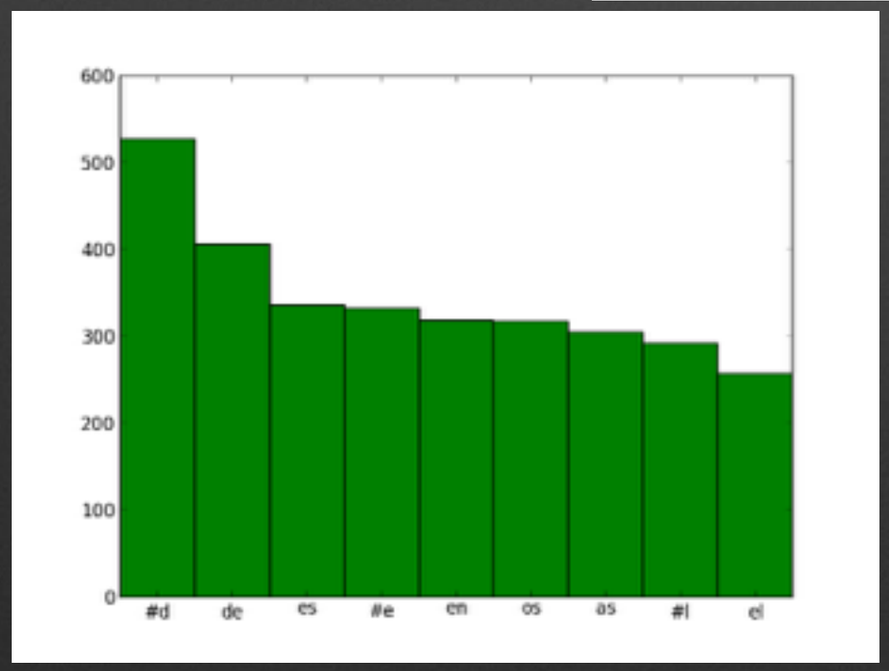
Classification

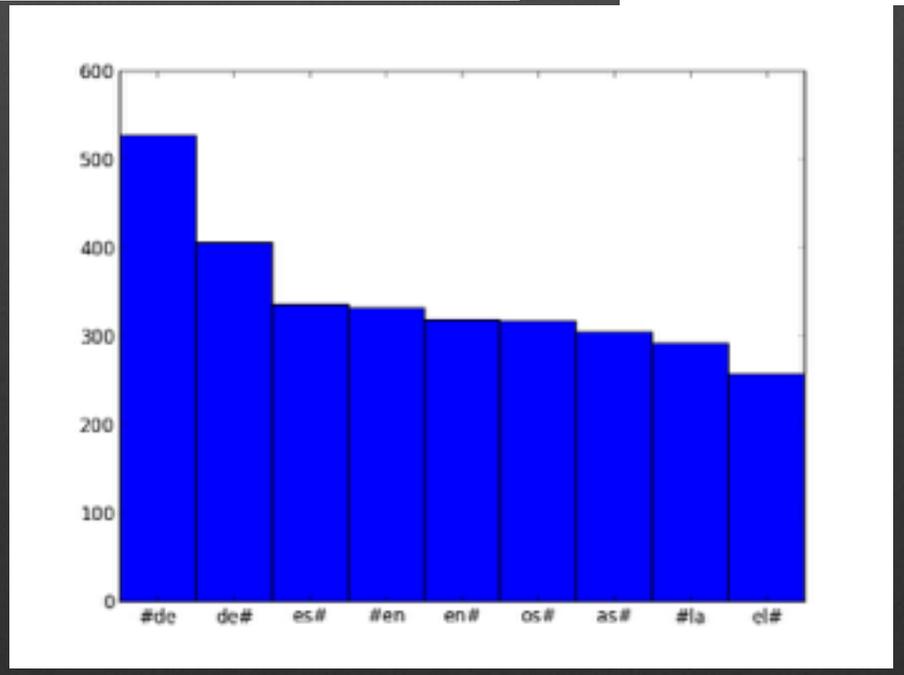
- N-gram models:

English ->









<- Spanish

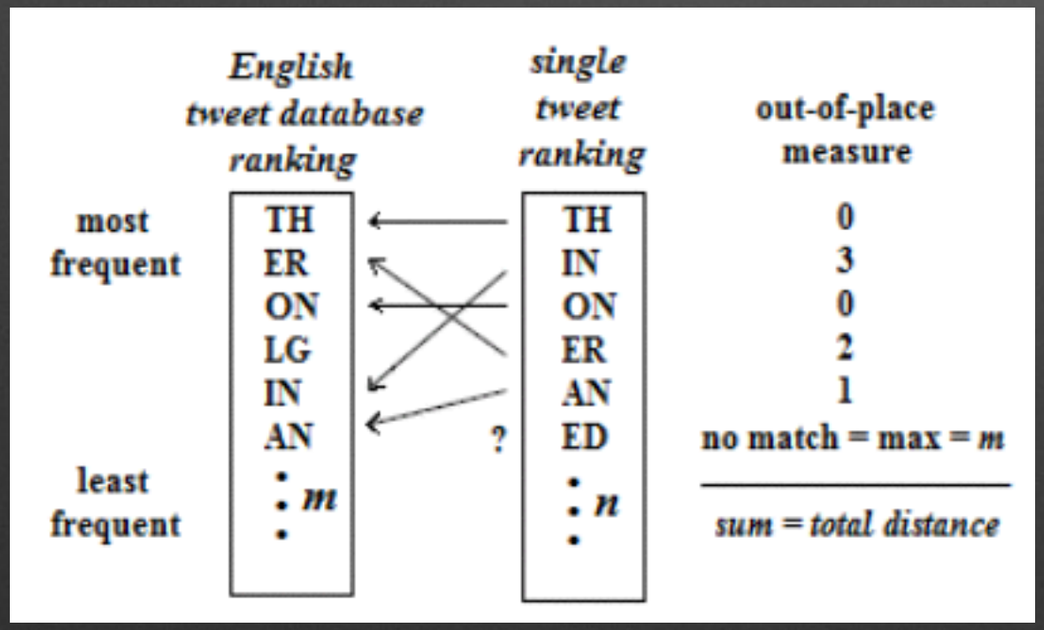
Classification

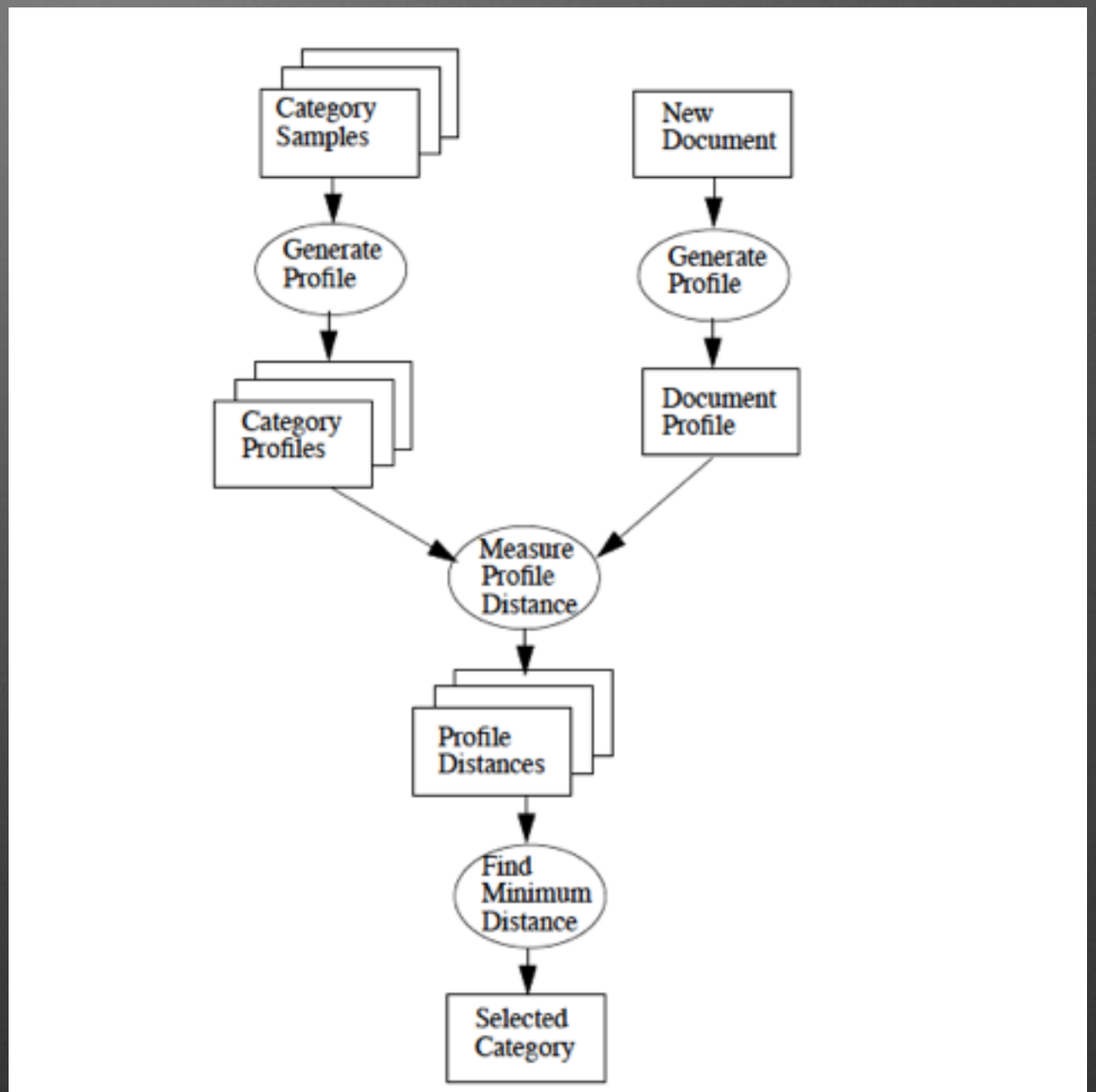
- N-gram models
  - Probability of a sentence belonging to a language.
  - Problem: N-grams that are not in our dataset make probability of the sentence 0.
- Solution: Lidstone smoothing

$$P_{LID}(X) = \frac{count(X) + \lambda}{N + B\lambda}$$

Classification

- N-gram models
  - Ranking models





Experiments

- Lidstone vs Ranking

	λ				
	0.1	0.3	0.5	0.7	1
Error	0.40625	0.490625	0.528125	0.56875	0.60625

	m				
		20	50	80	110
n	50	0.33125	0.280898876404	0.274774774775	0.345945945946
	80	0.331168831169	0.209302325581	0.224299065421	0.134831460674
	110	0.32	0.322580645161	0.346153846154	0.325581395349
	150	0.416666666667	0.3	0.32	0.333333333333

Conclusions

- Our approach successfully classified different languages, with high precision.
- As for the ranking methods, the voting method proved to be much better.
- Ranking methods proved to be more efficient and less computationally expensive.

Future Work

- Weighted voting could be implemented for the ranking method, in order to generate more accurate results.
- Take into account the character encoding to have more precision.
- Extend the language scope to a wider set.

Questions?