

A Simple Approach for Author Profiling in MapReduce

Suraj Maharjan, Prasha Shrestha, and Tamar Solorio

Introduction

- Task
 - Given an anonymous document
 - Predict
 - Age group [18-24 | 25-34 | 35-49 | 50-64 | 65-plus]
 - Gender [Male | Female]
 - Provided: Training data in English and Spanish
 - English – Blog, Reviews, Social Media, and Twitter
 - Spanish – Bog, Social Media, and Twitter

Motivation

- Started experimenting with PAN'13 data
- PAN'13 dataset
 - 1.8 GB of training data for English
 - 384 MB of training data for Spanish
- Explored MapReduce for fast processing of huge amount of data

Data Distribution

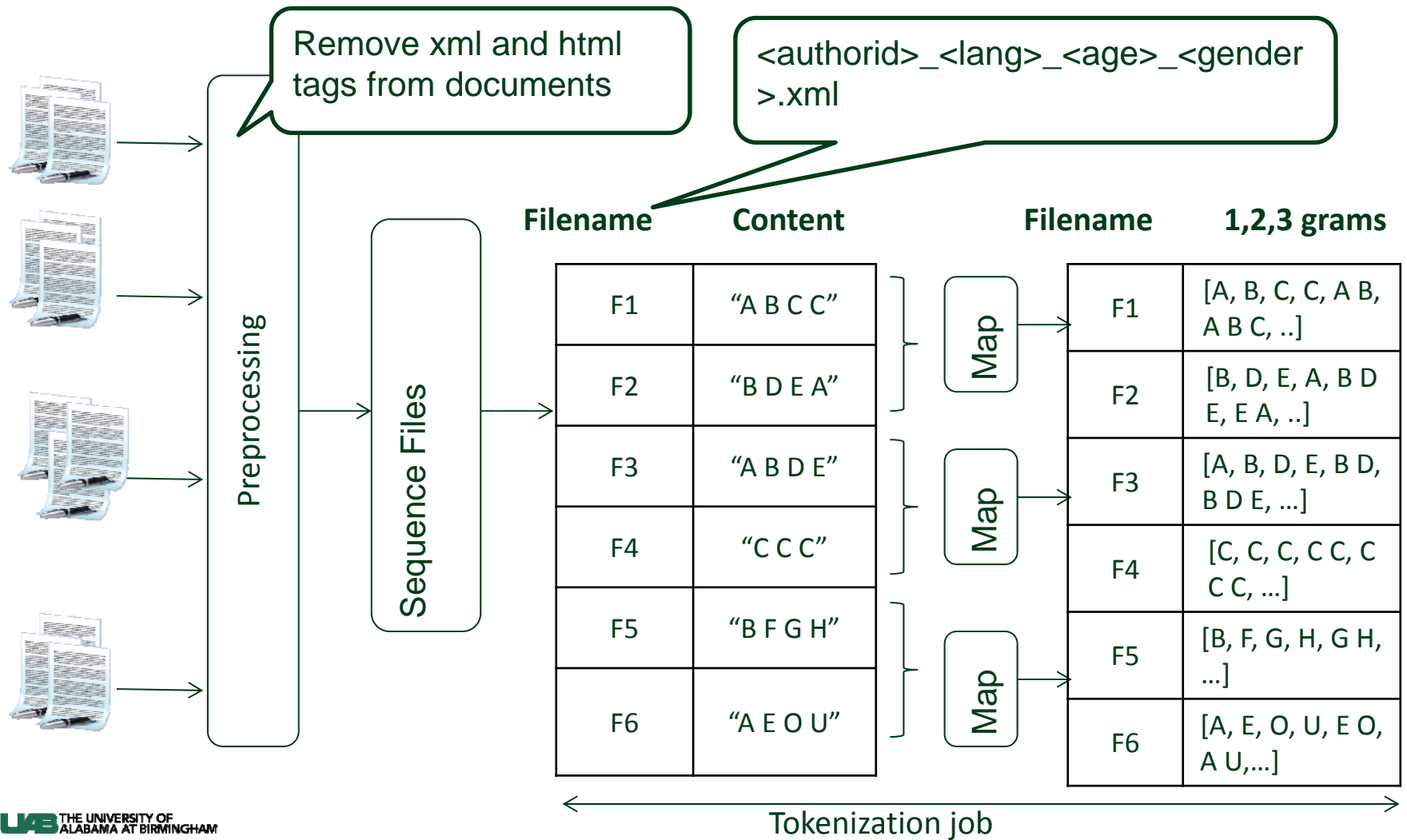
Category	English		Spanish	
	Files	Size (MB)	Files	Size (MB)
Blog	147	7.6	88	8.3
Reviews	4160	18.3	-	-
Social Media	7746	562.3	1272	51.9
Twitter	306	104.0	178	85.0
Total	12359	692.2	1538	145.2

Table 1: Training data distribution.

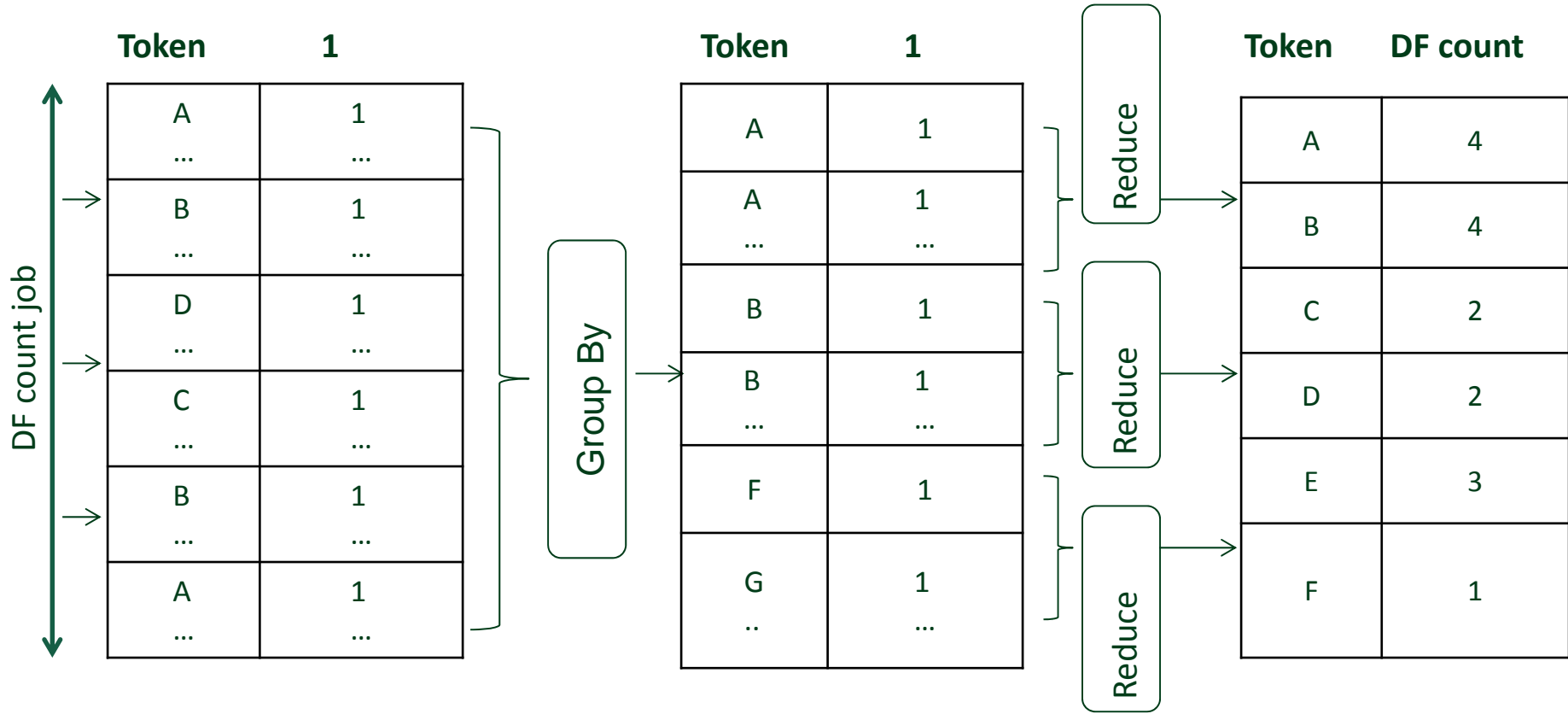
Methodology

- Preprocessing
 - Sequence File Creation
 - Tokenization
 - DF Calculation
 - Filter
- Features
 - Word n-grams (unigrams, bigrams, trigrams)
 - Weighing Scheme: TF-IDF
- Classification Algorithm
 - Logistic Regression with L2 norm regularization

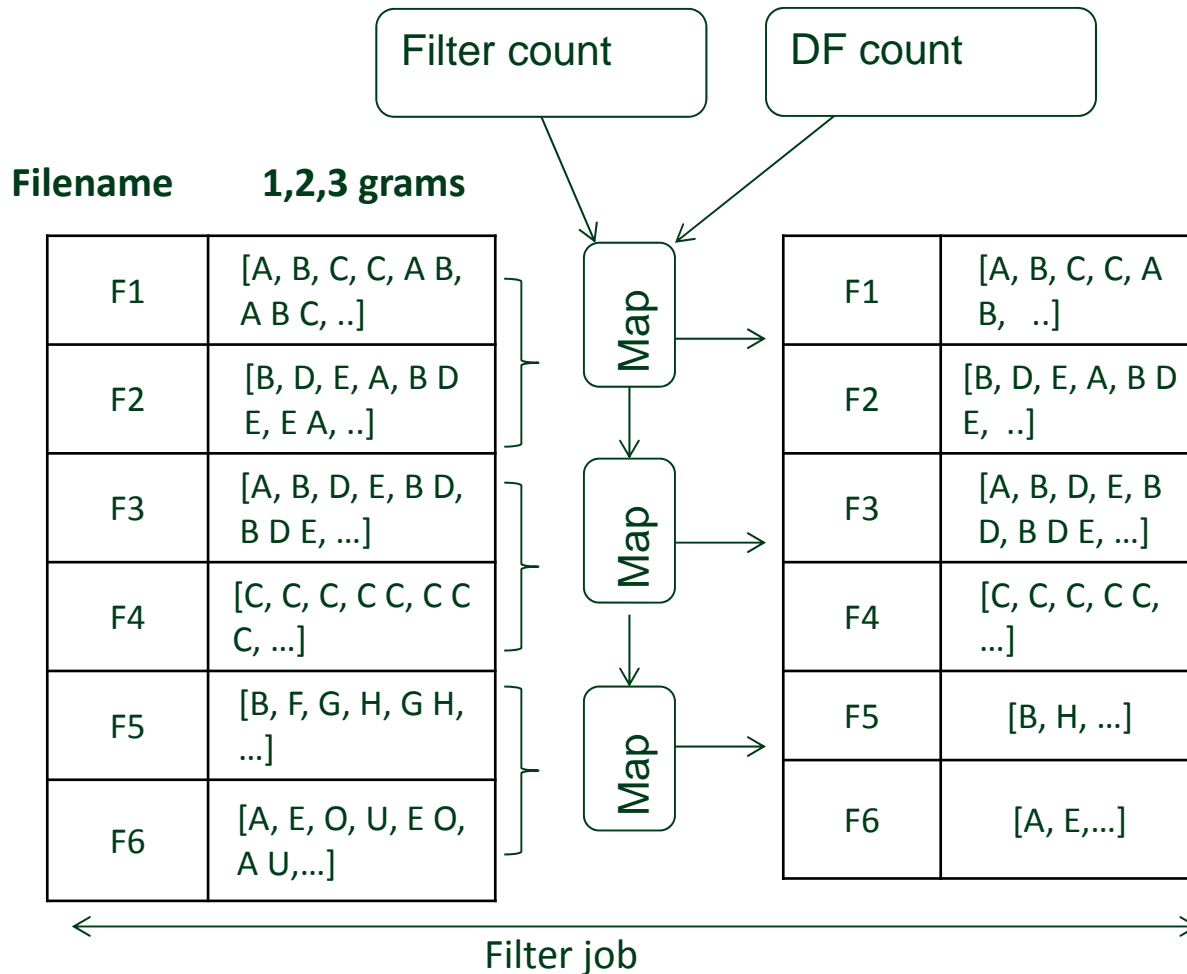
Tokenization



DF Calculation Job



Filter Job



TF-IDF Job

- Mapper
 - Setup:
 - Read in dictionary and DF score files
 - Map:
 - Map("<authorid>_<lang>_<age>_<gender>.xml", filtered token list)->("<authorid>_<lang>_<age>_<gender>.xml", VectorWritable)
 - Compute tf-idf scores for each token
 - Creates RandomSparseVector(mahout-math)
 - Finally writes vectors

Training

- Trained on :
 - Naïve Bayes (MR)
 - Cosine Similarity (MR)
 - Weighted Cosine Similarity (MR)
 - Logistic regression (LibLinear)
 - SVM (LibLinear)
- Final model uses LibLinear's logistic regression

Experiments

- Local Hadoop cluster with 1 master node and 7 slave nodes
- Each node has 16 cores and 12 GB memory
- Training data split into 70:30 ratio for training and development
- Modeled as 10 class classification problem

Experiments

Classification Algorithm	English (%)				Spanish (%)		
	Blogs	Reviews	Social Media	Twitter	Blog	Social Media	Twitter
Naïve Bayes	27.50	21.55	20.62	28.89	55.00	20.48	34.78
Cosine similarity	20.00	23.64	19.72	27.78	35.00	26.33	36.96
Weighted Cosine Similarity	30.00	23.16	19.97	26.67	40.00	22.07	32.61
Logistic Regression	27.50	23.08	20.62	33.33	35.00	25.80	32.61
SVM	25.00	22.28	19.80	32.22	30.00	26.33	34.78

Table 2: Accuracy for word 1, 2, 3 -grams for cross validation dataset.

Classification Algorithm	English (%)				Spanish (%)		
	Blogs	Reviews	Social Media	Twitter	Blog	Social Media	Twitter
Naïve Bayes	25.00	18.99	18.33	24.44	40.00	19.68	23.91
Cosine similarity	20.00	21.63	17.90	30.00	50.00	21.81	26.09
Weighted Cosine Similarity	20.00	21.15	16.78	23.33	40.00	19.68	28.26
Logistic Regression	22.50	21.71	16.78	25.56	35.00	23.67	17.39
SVM	20.00	20.83	15.92	24.44	35.00	23.14	17.39

Table 3: Accuracy for character 2, 3 -grams for cross validation dataset.

Experiments

- Separate Model: Different models for blog, social media, twitter and reviews per language
- Single Model: A single, combined model for each language

Classification Algorithm	English (%)		Spanish (%)	
	Separate Models	Single Model	Separate Models	Single Model
Naïve Bayes	21.21	20.13	23.53	21.04
Cosine similarity	19.89	17.34	27.83	27.60
Weighted Cosine Similarity	21.32	18.18	23.98	24.89
Logistic Regression	21.83	21.92	26.92	28.96
SVM	20.99	20.48	27.37	28.05

Table 4: Accuracy for single and separate models for all categories.

Results

- Number of features in English : 7,299,609
- Number of features in Spanish: 1,154,270

System	Average Accuracy(%)
PAN'14 Best	28.95
Ours	27.60
Baseline	14.04

Table 5: Accuracy comparison with other systems.

Results

Language	Category	Test 1				Test 2			
		Both	Age	Gender	Runtimes	Both	Age	Gender	Runtime
English	Blog	16.67	25.00	54.17	00:01:50	23.08	38.46	57.69	0:01:56
	Reviews	20.12	28.05	62.80	00:01:46	22.23	33.31	66.87	0:02:13
	Social Media	20.09	36.27	53.32	00:07:18	20.62	36.52	53.82	0:26:31
	Twitter	40.00	43.33	73.33	00:02:01	30.52	44.16	66.88	0:02:31
Spanish	Blog	28.57	42.86	57.14	00:00:35	25.00	46.43	42.86	0:00:39
	Social Media	30.33	40.16	68.03	00:01:13	28.45	42.76	64.49	0:03:26
	Twitter	61.54	69.23	88.46	00:00:43	43.33	61.11	65.56	0:01:10

Table 6: Accuracy by category and language on test dataset.

1st	2nd	3rd
-----	-----	-----

Conclusion

- Word n-grams proved to be better features than character n-grams for this task
- MapReduce is ideal for feature extraction from large dataset
- Our system works better when there is a large dataset
- Simple approaches can work

Demo

- <http://coral-projects.cis.uab.edu:8080/authorprofile14/>

Thank you.