LANGUAGE IDENTIFICATION USING G-LDA

Shubham Saini[1], Bhavesh Kasliwal[2], Shraey Bhatia[3]

[1, 2, 3]*Student, School of Computing Science and Engineering, Vellore Institute of Technology, India,*
shubham.saini2010@vit.ac.in, bhavesh.kasliwal2010@vit.ac.in, shraey.bhatia2010@vit.ac.in

Abstract

Language Identification has an important role in Natural Language processing applications as one of the pre-processing steps. There are various mechanisms in use today to achieve this task with brilliant recognition rates.

Recent years have seen rapid growth in international communication which has lead to the requirement of systems capable of correctly identifying languages of documents. Possible applications of language identification include information retrieval, web crawlers, text mining and email filtering.

The paper uses a process called G-LDA [1], which takes concepts from Latent Dirichlet Allocation (LDA) and Genetic Evolution techniques. This involves framing a set of words having a high frequency of occurrence in any given document. The method was tested on Leipzig Corpora. The phrases that were evolved through the generations reflected significant improvement.

Keywords: *Language Identification, Latent Dirichlet Allocation, Gibbs Sampling, Genetic Algorithm, Topic Modeling, Breeding, Fitness, Roulette Wheel.*

----------***----------

1. INTRODUCTION

Language Identification is the problem of automatically identifying the language a document is written in through the use of computer. Language Identification has an important role in Natural Language processing applications as one of the pre-processing steps.

Recent years have seen rapid growth in international communication which has lead to the requirement of systems capable of correctly identifying languages of documents. Also, with the spread of world-wide Internet access, text is available in a great number of languages other than English. Automatic processing of these texts for purposes like web crawling, reading aids, indexing etc. need a preliminary identification of the language used. Likewise, any system involving dictionary access must identify language before applying any language-specific operations.

The paper uses a process called G-LDA, which takes concepts from Latent Dirichlet Allocation [2] (LDA) and Genetic Evolution techniques. This involves framing a set of words having a high frequency of occurrence in any given document. The method was tested on Leipzig Corpora. The phrases that were evolved through the generations reflected significant improvement.

LDA is a way of automatically discovering topics within a documents collection. Each word in the documents collection will belong to a particular topic. The technique can be applied to a language identification corpus. The result will be a set of words having a high probability of occurrence within any given document.

Further, set of words generated earlier are subjected to genetic evolution techniques. This is done to get an improved set of words with a higher probability of occurrence within a document.

It was found that the method worked well without any pre-processing like tokenization, unlike the N-Gram based approach [7], and can handle e.g. Arabic documents and documents encoded in Unicode.

Also, the method is very easy to understand and implement, unlike the PPM approach [8], with almost similar detection rates.

2. SET GENERATION

A sub-set of sentences from Leipzig Corpora was taken. The languages chosen were Arabic, English, Guajarati, Hindi and Italian. All the sentences from these five languages were combined into a single document and sets of words from these documents were generated using the JGibbLDA [3] package. JGibbLDA is a Java implementation of LDA using Gibbs Sampling technique. A total of 5 sets with 200 words each were generated, each set representing a single language.

Now, for each word in each language set, the number of lines in the training data for that language matched, (N_e) was found. Genetic evolution techniques were applied on the words with $N_e > 0$.

3. GENETIC EVOLUTION ALGORITHM

Genetic algorithm was applied as follows:

- Loop for n generations
  - Find Fitness Function for each phrase
  - Breeding using roulette wheel selection
  - Filtering to produce a new generation

3.1 Fitness Function

Based on the number of lines matched, a fitness value is assigned to each phrase:

$$\text{Fitness} = \text{No. sentences containing the phrase} * \left(1+\frac{N-N_w}{N}\right)$$

Where
N = Threshold value
N_w = No. of words comprising the phrase

Threshold value is taken in order to provide a negative weight. Any phrase containing more than N words will receive fitness penalty, and phrases with less than N words receive a bonus.

3.2 Breeding

The process used created 1 child from 2 parents using the OR Method. Each survivor from previous generation breeds with a 2nd parent selected using Roulette Wheel Selection.

The OR Method – Take 2 parents, such as (Lottery|Cash) and (Cash|Win), breed them to produce the child (Lottery|Cash|Win). Here, $N_w = 3$ as the phrase comprises of 3 words.

Roulette Wheel Selection – It is a method for choosing a phrase from the breeding pool. Phrases with better fitness are more likely to be chosen.

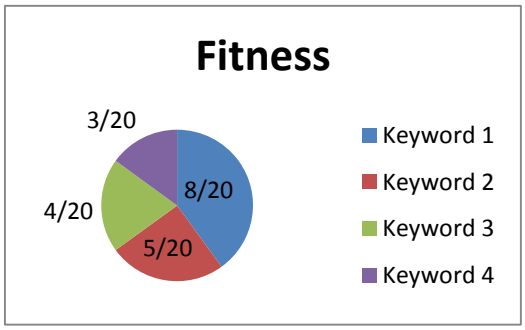


Chart -1: Roulette Wheel Selection

In Chart 1, the breeding pool consists of 4 phrases. The probability of Phrase 1 being selected is 8/20, the highest among all the phrases. This is based on the fact that Phrase 1 has the highest fitness of 8.

Breading is carried out as follows to produce new children

- Mark all Phrases as 'new'
- Loop till no phrase is marked 'new'
  - Take a phrase (P1) with highest fitness which is marked 'new'.
  - Mark P1 as 'old'
  - Take another phrase (P2) using Roulette Wheel Selection
  - Mark P2 as 'old'
  - Breed P1 and P2 to produce a child C = (P1|P2).
  - Mark C as 'old'

3.3 Filtering

Once breeding is done, fitness of the newly generated children is calculated and then filtering is performed in order to remove low fitness words. The words remaining after filtering constitute the new generation.

Filtering criteria may vary with need and situation. For the purpose of language identification, it was intended to keep only top 200 phrases. Hence, after each round of breeding, top 200 phrases were sent to the next generation and the rest of the words were discarded.

4. PROOF OF CONCEPT

In the Fitness function, as the threshold value (N) increases, the time required to find the number of sentences for each phrase increases with each generation as phrases with more number of words tend to survive. After performing several tests, the threshold value was chosen to be 5. This is the optimum value which balances the time required for finding the number of sentences for each phrase and the number of new generations required. The breeding is stopped when the increase in the fitness of the phrases become insignificant.

The top 10 phrases of the first two generations for the five languages, with their fitness score is given below:

Table -1: Top 10 Arabic language phrases with fitness values

Generation 0		Generation 1	
و	124.7579956	و	124.758
في	101.0699997	و;فقد	113.152
من	87.17399597	في	101.07
على	54.66600037	في;الولايات المتحدة	96.81601
أن	47.12400055	على;أن	90.48

ان	38.14199829	من	87.174
عن	33.28199768	من;دول	83.888
مع	30.23999977	على	54.666
◻تي	27.395998	عن◻ى	52.512
◻ى	25.79399872	مع;لا	49.504

Table -2: Top 10 English language phrases with fitness values

Generation 0		Generation 1	
an	91.27799988	year;named	145.474
of	85.03200531	a;had	140.576
and	80.85599518	However;Last	140.182
to	73.9980011	the;two	117.312
be	39.94200134	and;with	94.496
is	36.45000076	an	91.278
for	33.13799667	an;do	90.35201
was	31.12199974	of	85.03201
de	27.59399986	to;so	82.96
with	25.45199966	and	80.856

Table -3: Top 10 Guajarati language phrases with fitness values

Generation 0		Generation 1	
છે	80.7480011	છે;અને	232.33
જ	80.40599823	કે;હતી	180.068
આ	77.45400238	આ;છે.	126.192
ન	64.80000305	છે;એ	112.64
છે.	64.51199341	જ;કરવા	84.81601
એ	45.97199631	છે	80.748
તે	40.12199783	છે	80.748
અને	37.96199799	જ	80.406
કે	35.7840004	આ	77.454
પણ	26.51399994	ન	64.8

Table -4: Top 10 Hindi language phrases with fitness values

Generation 0		Generation 1	
के	98.9280014	है;प्रति	216.79
न	97.91999817	हैं;उस	170.534
है	93.31200409	कहा;उन्हें	159.642
में	87.01200104	में;की	140.56
की	71.11799622	न;हो	116.704
व	67.98599243	को;का	108.96
को	62.22599792	के	98.928
का	60.35400009	न	97.92
कि	59.84999847	है	93.312

Table -5: Top 10 Italian language phrases with fitness values

Generation 0		Generation 1	
di	105.1920013	che;a	161.056
in	73.56599426	anni;uno	147.448
del	72.46799469	di;ha	124.768
che	51.02999878	di	105.192
da	44.51399994	in;da	104.96
su	36.18000031	in	73.56599
o	36.16199875	del;l	73.296
ha	35.17200089	del	72.46799
e	34.0019989	su;della	57.168
è	34.0019989	che	51.03

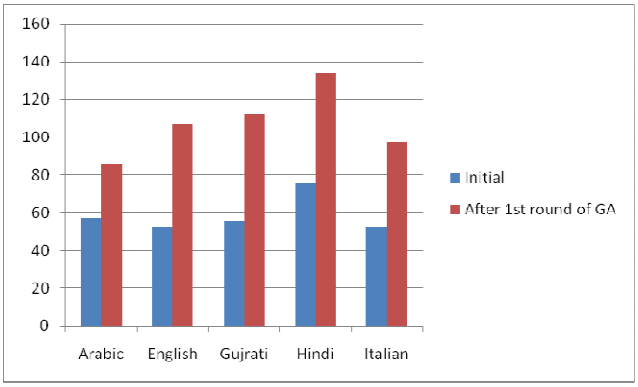


Chart -2: Average Fitness of Top 10 phrases of each language

It is evident from Chart 2 that average fitness of the phrases increases after applying the genetic algorithm.

To check the effectiveness of the evolved phrases, the following method was adopted. The algorithm returns the language of a document D:

- For Each Language L_i
  - For each phrase $K_i \in L_i$

- $C(K_i)$ = No. of lines of D containing K_i
- $S(L_i)=\sum C(K_i)$

- Return L, where S(L) = Max(S(Li))

This technique was applied to 15 web-pages of the 5 training languages (Arabic, English, Gujarati, Hindi and Italian). The system was able to identify the language for every web page correctly (giving 100% recognition rate).

It maybe be noted that the technique was applied to moderate (≥1000 words) sized web-pages. The identification rate may decrease when applied to very small sized documents (<100 words). However, this can be improved by taking a bigger training data, and retaining more words after each breeding stage.

CONCLUSIONS

Combining the Genetic evolution technique, which simulates processes of nature, with topic modeling method, the set of phrases generated effectively identify the language of a document.

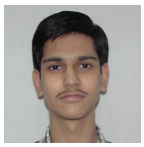
This technique, when applied to 15 web-pages of the 5 training languages (Arabic, English, Gujarati, Hindi and Italian), was able to identify the language for every web page correctly (giving 100% recognition rate).

REFERENCES

[1]. Shubham Saini, Bhavesh Kasliwal and Shraey Bhatia. "Spam Detection using G-LDA" International Journal of Advanced Research in Computer Science and Software Engineering 3.10 (2013).
[2]. Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." the Journal of machine Learning research 3 (2003): 993-1022.J. Breckling, Ed., The Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.
[3]. Phan, Xuan-Hieu, and Cam-Tu Nguyen. "Jgibblda: A java implementation of latent dirichlet allocation (lda) using gibbs sampling for parameter estimation and inference." (2006).
[4]. Ramage, D., and E. Rosen. "Stanford topic modeling toolbox." (2011).
[5]. Wall, Matthew. "GAlib: A C++ library of genetic algorithm components."Mechanical Engineering Department, Massachusetts Institute of Technology 87 (1996): 54.
[6]. Kranig, Simon. "Evaluation of language identification methods." Bakalárska práca, Universität Tübingen, Nemecko (2005)
[7]. Cavnar, William B., and John M. Trenkle. "N-gram-based text categorization." Ann Arbor MI 48113.2 (1994): 161-175.
[8]. Teahan, William J., and David J. Harper. "Using compression-based language models for text categorization." Language Modeling for Information Retrieval. Springer Netherlands, 2003. 141-165

BIOGRAPHIES

Shubham Saini is a senior year undergraduate student majoring in Computer Science and Engineering. His research interests are Information Retrieval, Semantic Web and Knowledge Discovery in Databases.



Bhavesh Kasliwal is a senior year undergraduate student majoring in Computer Science and Engineering. His research interests are Algorithm Design and Analysis, Data Analytics.

Shraey Bhatia is a senior year undergraduate student majoring in Computer Science and Engineering. His research interests are Information Security and Cryptography.