Unstructured Text Analysis

Aalap Jethwa, Ankit Singh
Guided by :Prof. Uttam Chauhan
Computer Engineering, Vishwakarma Government Engineering College Chandkheda

Introduction

The opportunity cost of any business to ignore unstructured data is paramount in today's fierce competitive world. According to an IDC survey, unstructured data takes a lion's share in digital space and approximately occupies 80% by volume compared to only 20 for structured data.

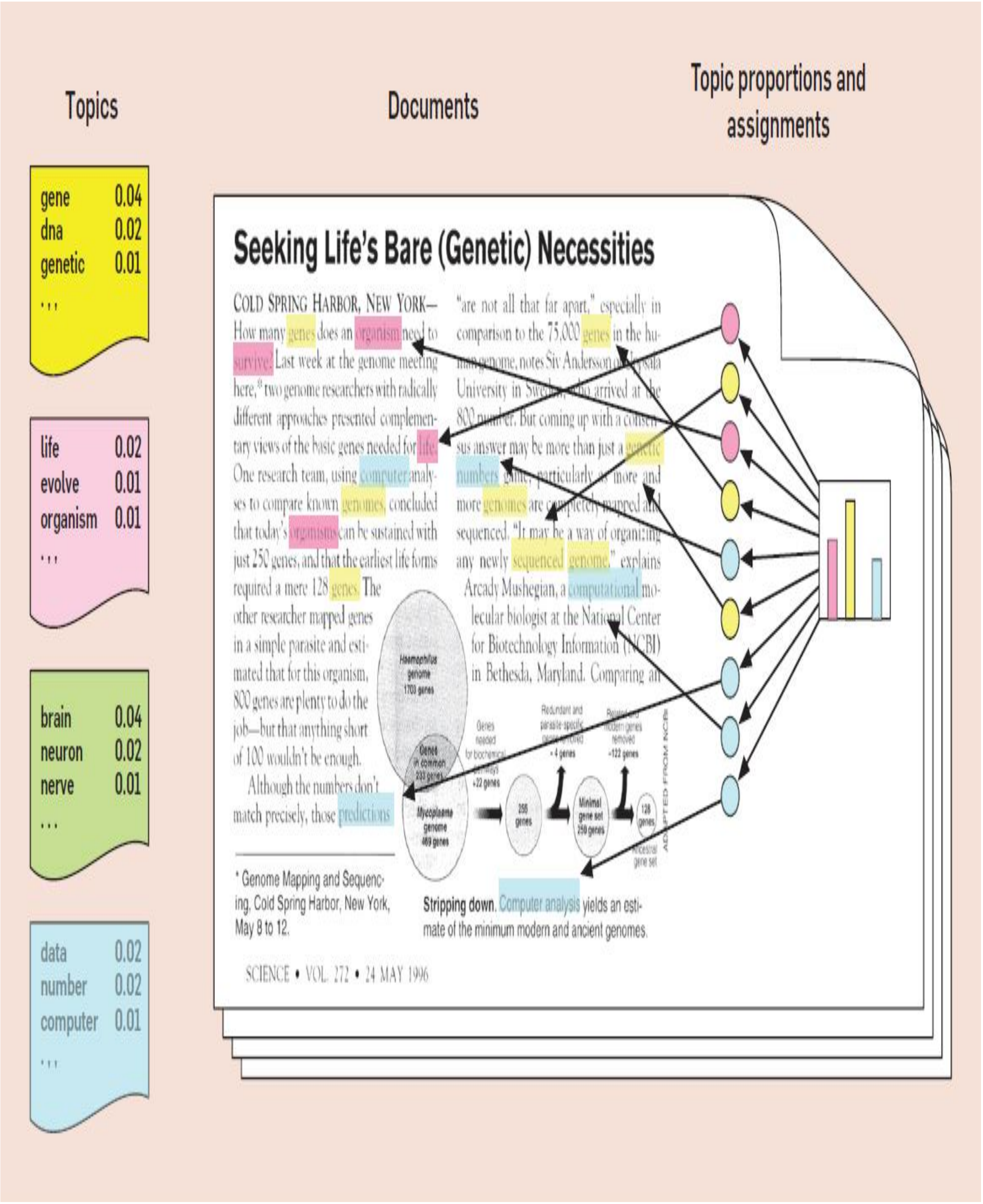
While the unstructured data is available in abundance, the number of software products and solutions that can accurately analyze the text, present insights in an understandable manner along with the ability to integrate such insights readily into other extant models that use numerical only data are rare.

A lot of the challenges in this space arise from the fact that natural language provides the flexibility to convey exactly the same meaning in umpteen different ways, or worse, exactly the same statement in a different context may convey completely different meaning. Machine learning algorithms designed to analyze numerical data exactly know the structure of numbers and they are pre-programmed to process the data with precision. In case of natural language, it gets very problematic.

Applications

- Massive automatic movies indexation from subtitles.
- Topical stock quote motions
- Behavior mining of Internet users.
- Expert AI Systems
- News Editors

Example



Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.

LDA assumes the following generative process for each document **w** in a corpus *D*:

1. Choose *N* Poisson(x).

2. Choose q Dir(a).

3. For each of the *N* words *wn*:

(a) Choose a topic *zn* Multinomial (q).

(b) Choose a word *wn* from *p*(*wn j zn;*b), a multinomial probability conditioned on the topic *zn*.

Given the parameters a and b, the joint distribution of a topic

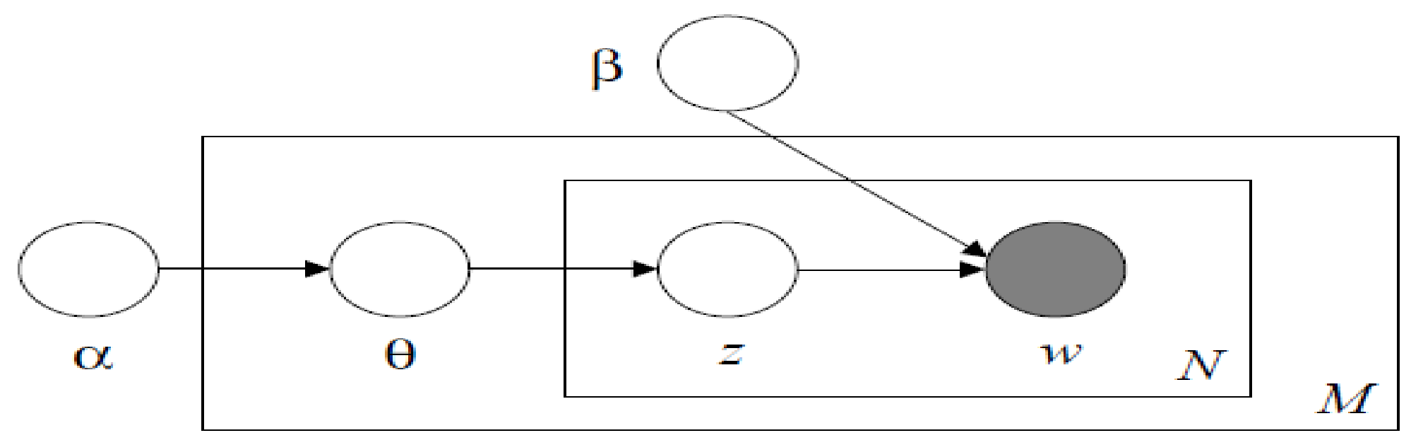


Figure-1 Plate Notation-LDA

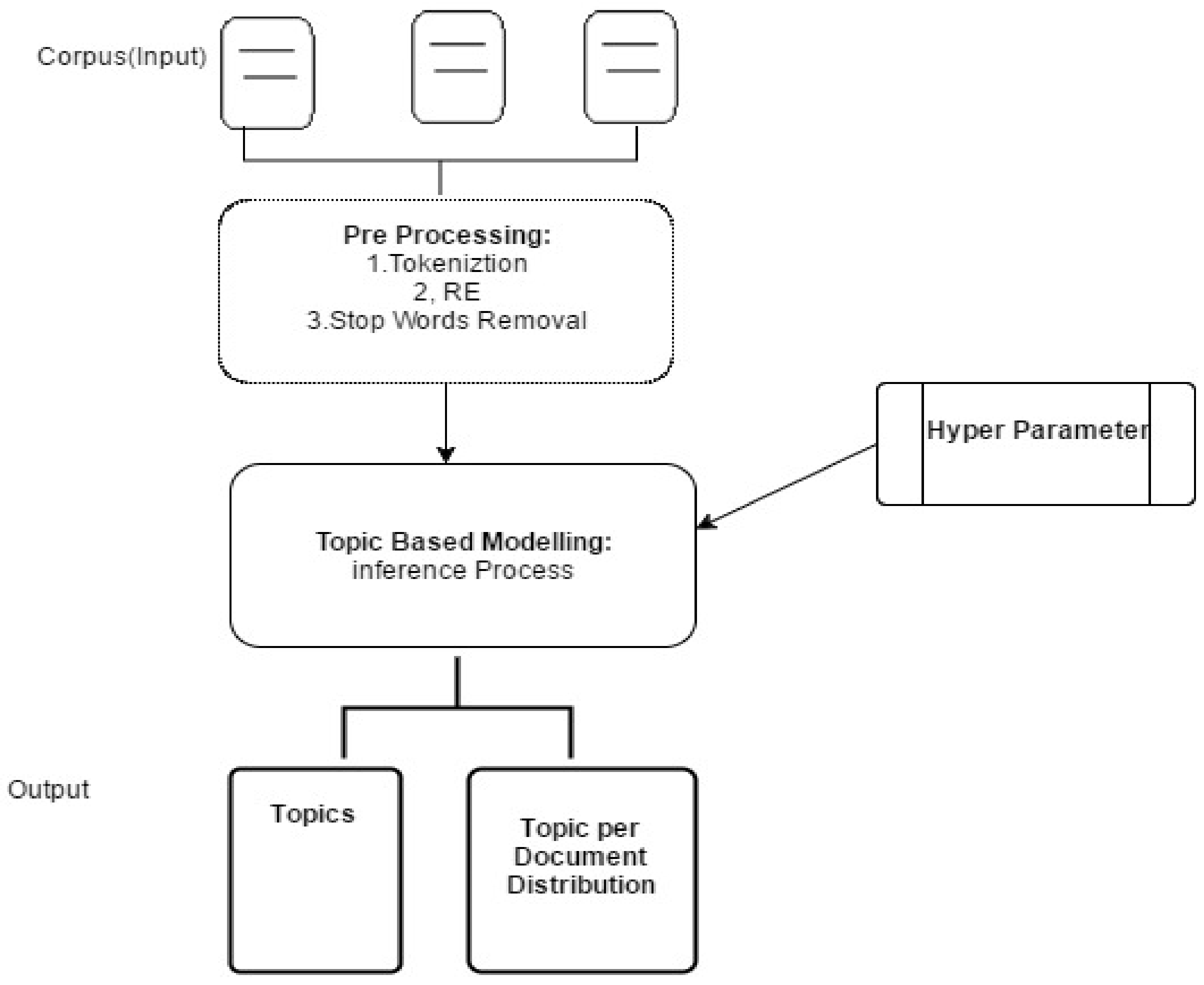
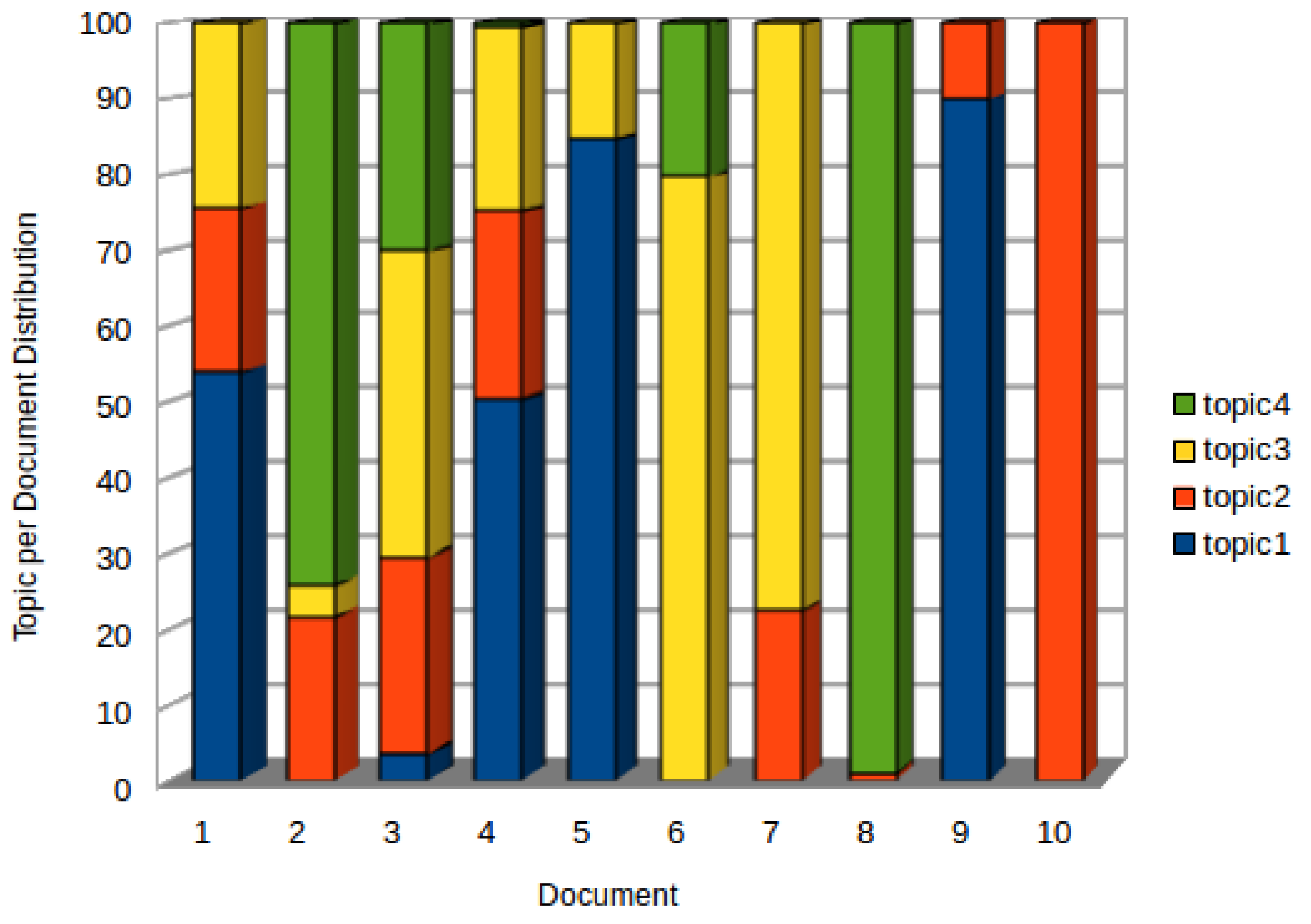


Figure-2 System Flow



Gibbs Sampling

Gibbs sampling or a Gibbs sampler is a Markov chain Monte Carlo (MCMC) algorithm for obtaining a sequence of observations which are approximated from a specified multivariate probabilit distribution, when direct sampling is difficult. This sequence can be used to approximate the joint distribution to approximate the marginal distribution of one of the variables, or some subset of the variables (for example, the unknown parameters or latent variables); or to compute an integral (such as the expected value of one of the variables). Typically, some of the variables correspond to observations whose values are known, and hence do not need to be sampled.

Gibbs sampling is commonly used as a means of statistical inference, especially Bayesian inference. It is a randomized algorithm i.e. an algorithm that makes use of random numbers.

TF: Term Frequency.

TF (t) = (Number of times term t appears in a document) / (Total number of terms in the document).

IDF: Inverse Document Frequency.

IDF (t) = log_e (Total number of documents / Number of documents with term t in it).

```
Algorithm 1 Gibbs sampler
Initialize $x^{(0)} \sim q(x)$
for iteration $i = 1, 2, \ldots$ do
    $x_1^{(i)} \sim p(X_1 = x_1 | X_2 = x_2^{(i-1)}, X_3 = x_3^{(i-1)}, \ldots, X_D = x_D^{(i-1)})$
    $x_2^{(i)} \sim p(X_2 = x_2 | X_1 = x_1^{(i)}, X_3 = x_3^{(i-1)}, \ldots, X_D = x_D^{(i-1)})$
    $\vdots$
    $x_D^{(i)} \sim p(X_D = x_D | X_1 = x_1^{(i)}, X_2 = x_2^{(i)}, \ldots, X_D = x_{D-1}^{(i)})$
end for
```

Literature

•We used LDA to represent document as random mixtures over latent topics , referred from Latent Dirichlet Allocation By *David M. Blei*, *Andrew* Y. *Ng* and Michael I. Jordan. University of California, Berkeley.

•Gibbs Sampling by Ilker Yildirim Department of Brain and Cognitive Sciences University of Rochester.

•Variational Dirichlet Process by David Blei and Micheal I Jordan of Columbia University.

•To implement all algorithms we preferred python and libraries like sdkit-learn for machine learning and pandas for data mining. From web site - https://docs.python.org .

Conclusion

With the three months research that we carried and by trial and error the models we implement we can state that the final model at which we arrived provides use the best result among all the models that we implemented till now. With the refinement and the tweaking a few parameter here and there we are expecting to even get more good result. With the adoption of Topic based modeling and ultimately shifting to distributed architecture is a good omen for our system that we can conclude by the amount of time required to process the result on a single machine. Finally to summary we can conclude that with some minor changes and a distributed approach we are bounded to achieve more accurate results.