A description...

A Project Report

ON

**TOPIC MODELLING ON INDIAN RESEARCH PAPERS**

BY

APOORVA K H-1PI13CS034

AARTI JIVRAJANI-1PI13CS002

DEBARATI DAS-1PI13CS052

Guide

Dr. KAVI MAHESH  
Dean of Research,   
PESIT-CSE  
Bangalore

January 2015-May 2015

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PES INSTITUTE OF TECHNOLOGY

A description...

PES Institute of Technology,

(An autonomous institute under VTU)

100 Feet Ring Road, BSK 3rd Stage, Bangalore-560085

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**CERTIFICATE**

Certified that the Special Topic: Mini Project work entitled **Topic Modelling On Indian Research Papers** is a bona fide work carried out by **Debarati Das, Apoorva KH** and **Aarti Jivrajani** in partial fulfilment for the award of degree of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum during the academic semester August 2014 to December 2014.

Signature of the Guide Signature of the HOD

(Dr. Kavi Mahesh) (Prof. Nitin V Pujari)

Apoorva KH -1PI13CS034

Aarti Jivrajani -1PI13CS002

Debarati Das -1PI13CS052

**Abstract**

Scientists need new tools to explore and browse large collections of scholarly literature. Thanks to organizations such as JSTOR, which scan and index the original bound archives of many journals, modern scientists can search digital libraries spanning hundreds of years.

A scientist, suddenly faced with access to millions of articles in her field, is not satisfied with simple search. Effectively using such collections requires interacting with them in a more structured way: finding articles similar to those of interest, and exploring the collection through the underlying topics that run through it.

The central problem is that this structure—the index of ideas contained in the articles and which other articles are about the same kinds of ideas—is not readily available in most modern collections, and the size and growth rate of these collections preclude us from building it by hand.To develop the necessary tools for exploring and browsing modern digital libraries, we require automated methods of organizing, managing, and delivering their contents.

Topic models are probabilistic models for uncovering the underlying semantic structure of a document collection based on a hierarchical Bayesian analysis of the original texts or using machine learning algorithms.

In this project we modelled the network of topics present in a corpus of research papers published in India in the year 2013. This modelling was done using two approaches, LDA (probabilistic) and NMF (machine learning approach). The final topic network was then visualised and made web-explorable.

**Acknowledgement**

We use this opportunity to express our gratitude to everyone who supported us throughout the course of this Special Topic project. We are thankful for their inspiring guidance, invaluable constructive criticism and friendly advice during the project work. We are sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

We express our warm thanks to Dr. Kavi Mahesh, our project Mentor for his constant support and guidance during our Work in the K.A.N.O.E Centre of PESIT. We would also like to thank Professor I.K.Ravichandra Rao and Professor KS Raghavan for their valuable inputs in our project and their continuous guidance.

Thanking you,

Apoorva KH

Aarti Jivrajani

Debarati Das

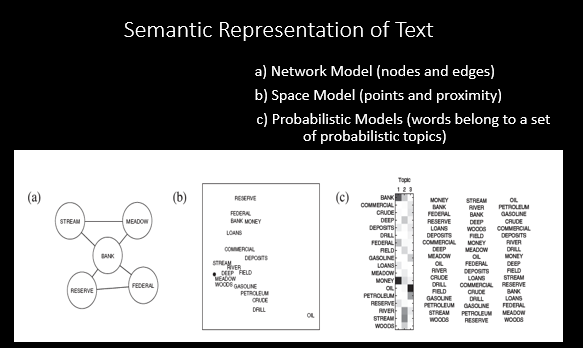
**Table of Contents**

|  |  |
| --- | --- |
| 1 | Introduction |
| 2 | Problem Definition |
| 3 | Literature Survey |
| 4 | System Requirement Definition |
| 5 | System Design |
| 6 | Algorithms Used |
| 7 | Result Analysis |
| 8 | Further Enhancements |
| 9 | Bibliography |

1. **INTRODUCTION**

Text mining, refers to the process of deriving high-quality information from text. High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning.

Text can be semantically represented in three ways as shown below:



Topic Modelling is a form of Text Mining, or a means of identifying pattern in a text.

**What is Topic Modelling?**

A topic, in this context, can be described as a recurring pattern of co-occurring words. A topic modelling tool usually looks through the corpus for such word-clusters and groups them by similarity. In a good topic model words in the topic make sense, E.g.: “human, DNA, sequence, genome” clearly are on a topic related to Genetics.

Topic Modelling is not necessarily useful as evidence but it makes an excellent tool for discovery.

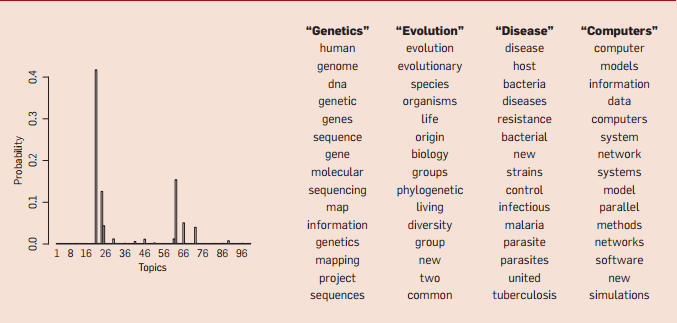


Figure 1 Topic Modelled Result

**What is the layman’s view of Topic Modelling?**

One way to think of Topic Modelling is to imagine working through an article with a set of highlighters. As you read through the article you use a different color for the keywords/themes within the paper as you come across them.

When you are finally done you can copy out the words as grouped by the color you assigned them. These lists of words constitute a topic and each color represents a different topic.

Topic modelling gives us a way to infer the latent structure behind a collection of documents. In principle, it could work at any scale, but we tend to think human beings are already pretty good at inferring the latent structure in (say) a single writer’s oeuvre. We suspect this technique becomes more useful as we move toward a scale that is too large to fit into human memory.

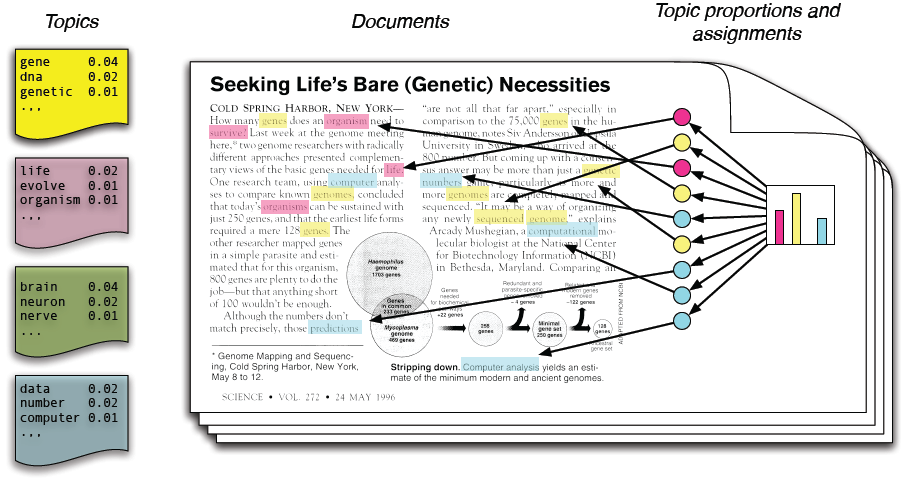


Figure 2 Topic Modelling- a Layman’s View

**How does Topic Modelling actually work?**

Actual topic modelling programs are determined by mathematics. Topic Modelling is an updated version of Latent Semantic analysis. It works great on large collections of well written content.

There are various methods to implement Topic Modelling. Latent Dirichlet allocation, or LDA, which seems to be the most widely used model in the humanities. LDA has strengths and weaknesses, and it may not be right for all projects.

Methods like NMF (Non-negative Matrix Factorisation) And K Means Clustering can also be used for topic modelling. In fact, it has very recently been recognised that NMF is a highly effective alternative to LDA.

In this project, we have used a combination of Unsupervised Machine Learning techniques (NMF), Probabilistic modelling techniques (LDA), Graph Theory and Social Network Analysis to model a research paper network and analysed it.

Figure 3 Triad of Techniques Used

1. **Problem definition**

The Problem statement is as follows:

* Given Indian Research Papers from the year 2013 as a dataset. Analyse this dataset using a methodology and come up with a graphical depiction of all the major topics that exist in the dataset.
* Use two different Topic modelling techniques and contrast them over the same dataset.
* Using Graphical Metrics depict which topics were in vogue during the year 2013.

1. **Literature survey**

We looked at the following Techniques and References to formulate the process for our project:

1. *Text Mining Techniques*

* Latent Dirichlet Allocation (LDA)
* Non negative Matrix Factorisation (NMF)

1. *Graph Theory*

Since social networks are represented by Graphs, we have looked at the Graph Theory concepts and Algorithms.

1. *Social Network Visualisation*

There are specific concepts and metrics of Graph theory that any social network analysis uses. We have looked at some the metrics that we can use in our analysis of the social network that we will construct.

1. **System requirements specification**

Text Mining and processing of the graph data are computationally intensive .If you’re working with extremely large file collections – or indeed, very large files – you may run into issues with your heap space, your computer’s working memory. This issue will initially arise during the import sequence, if it is relevant. By default, MALLET allows for 1GB of memory to be used. For our project, we have chosen a granularity at an appropriate level to suit our desktop environment. About 4GB of memory is more than sufficient.

1. **System design**

What do you need to perform topic modelling?

* **A corpus, preferably a large one**
* **A tool to do the topic modelling**. It is important to be aware that you need to train these tools. Topic modelling tools only return as many topics as you tell them to; it matters whether you specify 50, 5, or 500. If you imagine topic modelling as a switchboard, there are a large number of knobs and dials which can be adjusted. These have to be tuned, mostly through trial and error, before the results are useful.
* Topic modelling output is not entirely human readable. One way to understand what the program is telling you is through a visualization, but be sure that you know how to understand what the visualization is telling you.

For the current iteration of this project, there are three sub-parts of the project as follows:

* 1. **Text Processing**

With some tools, we had to prepare the corpus before we could topic model. Essentially we tokenized the text, changing it from human-readable sentences to a string of words by stripping out the punctuation and removing capitalization.

We can also tell it to ignore “stop-words” which you define, which usually include things like a, the, and, etc. What we (hopefully) ended up with was a document with no capitalization, punctuation, or numbers to throw off the algorithms.

There are a number of ways to clean up text for topic modelling (and text mining). We have used [Python and Regular Expressions](http://www.fredgibbs.net/clio3workspace/blog/cleaning-bad-ocr-with-regular-expressions-and-python/).

* 1. **Topic Extraction**

Topic modelling programs do not know anything about the meaning of the words in a text. Instead, they assume that any piece of text is composed (by an author) by selecting words from possible baskets of words where each basket corresponds to a topic.

If that is true, then it becomes possible to mathematically decompose a text into the probable baskets from whence the words first came. The program goes through this process over and over again until it settles on the most likely distribution of words into baskets, which we call topics.

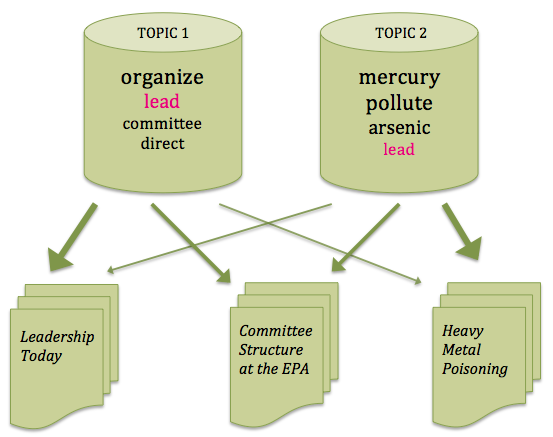
****

Figure 4 Topic Model buckets and their associated documents (Statistical Inference)

There are different approaches possible to model topics, in this project we have attempted two such methods.

* + 1. **Using LDA Approach ( Probabilistic Approach)**

*Latent Dirichlet allocation* (LDA), a generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modelled as a finite mixture over an underlying set of topics. Each topic is, in turn, modelled as an infinite mixture over an underlying set of topic probabilities. In the context of text modelling, the topic probabilities provide an explicit representation of a document. The algorithm ([called LDA](http://en.wikipedia.org/wiki/Latent_Dirichlet_allocation)) knows nothing “meta” about the articles (when they were published, say), and it knows nothing about the order of words in a given document.

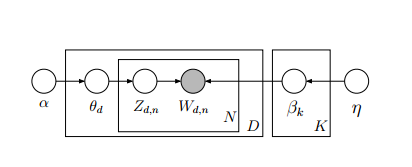


Figure 5 Pictorial Representation of LDA

Three latent variables are generated from LDA:

* + Topic distribution per document : P(z) = θ(d)
  + Word distribution per topic: P(w, z) = φ(z)
  + Word-Topic assignment: P(z|w)

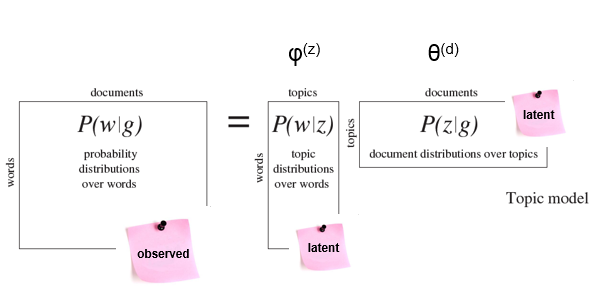


Figure 6 LDA Latent Variables

The algorithm involves asking two essential questions.

* **How much does the Document like the Topic?**
* **How much does the Topic like the Word?**

LDA represents documents as **mixtures of topics** that spit out words with certain probabilities. It assumes that documents are produced in the following fashion, that when writing each document, you:

* Decide on the number of words N the document will have (say, according to a Poisson distribution).
* Choose a topic mixture for the document (according to a Dirichlet distribution over a fixed set of K topics). For example, assuming that we have the two food and drinks topics, you might choose the document to consist of 1/3 food and 2/3 drinks.
* Generate each word in the document by:
* First picking a topic (according to the multinomial distribution that you sampled above; for example, you might pick the food topic with 1/3 probability and the drinks topic with 2/3 probability).
* Then using the topic to generate the word itself (according to the topic's multinomial distribution). For instance, the food topic might output the word "broccoli" with 30% probability, "bananas" with 15% probability, and so on.

Assuming this generative model for a collection of documents, LDA then tries to backtrack from the documents to find a set of topics that are likely to have generated the collection.

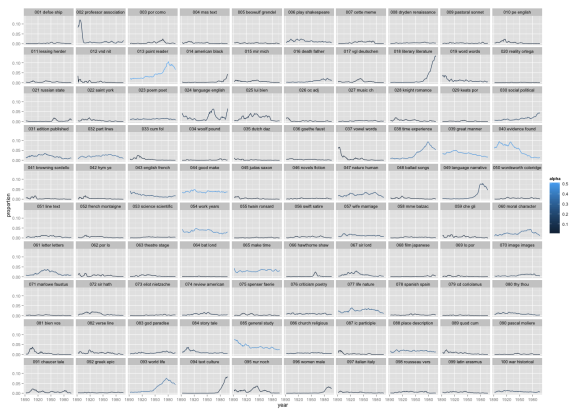
****

Figure 7 Presence of "Topic" distributions in a Document

* + 1. **Using NMF Approach**

NMF (Non-negative matrix factorization) is an unsupervised family of algorithms that simultaneously perform dimension reduction and clustering. NMF produces a “parts-based” decomposition of the latent relationships in a data matrix.

**Process:**

1. Construct vector space model for documents (after stop word filtering), resulting in a term-document matrix A. The document-term matrix **A** for this example is represented visually below as a heat map, where each row is a document, each term is a column, and a blue entry indicates that a term appears in a document at least once.

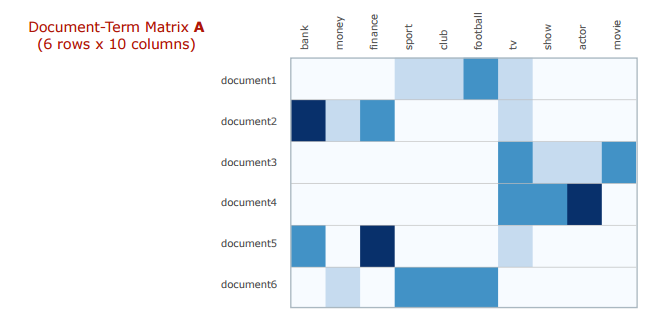
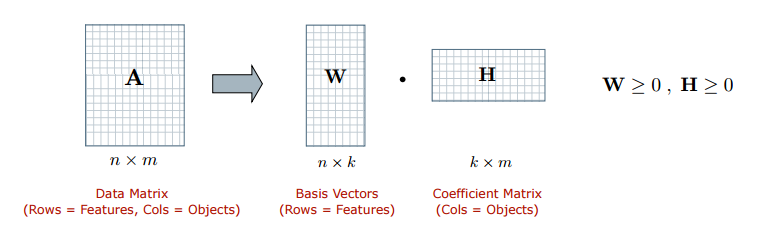


Figure 8 Heat map of Document-Term Matrix

1. Apply TF-IDF term weight normalisation to A.
2. Normalize TF-IDF vectors to unit length.
3. Initialise factors using NNDSVD (Nonnegative Double Singular Value Decomposition) on A.
4. Apply Projected Gradient NMF to A.

**Output Analysis of NMF**



NMF takes this non-negative matrix as an input, and factorizes it into two smaller non-negative matrices **W** and **H**, each having *k* dimensions.

When multiplied together, these factors approximate the original matrix **A.** The key user-specified parameter *k* controls the number of topics that will be produced.

The rows of the matrix **W** provides weights for the input documents relative to the *k* topics – these values indicate the strength of association between documents and topics. The columns of the matrix **W** provide weights for the terms relative to the topics. By ordering the values in a given column and selecting the top-ranked terms, we can produce a description of the corresponding topic.

The resulting *k* topics are defined by: (a) topic descriptions as given by the top-ranked terms in the columns of the factor **W**; (b) document membership weights as given by the values in the rows of **H**.

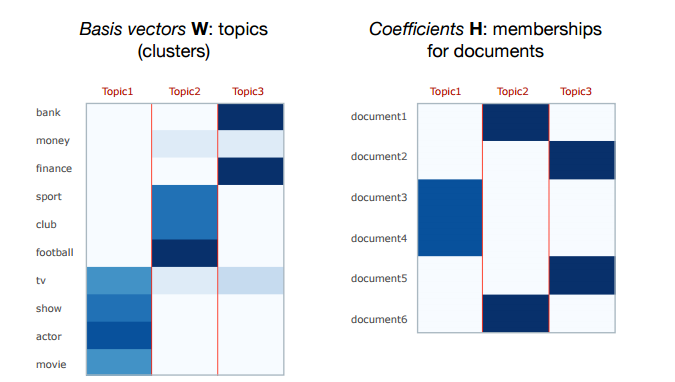


Figure 9 Output of NMF Topic Modelling

* 1. **Topic Network Construction**

Networkx (Python based open source library) is used to build the social network graph from weighted matrix. Gephi (an interactive visualization and exploration platform) is used to visualise the graph.

1. **Algorithms Used**

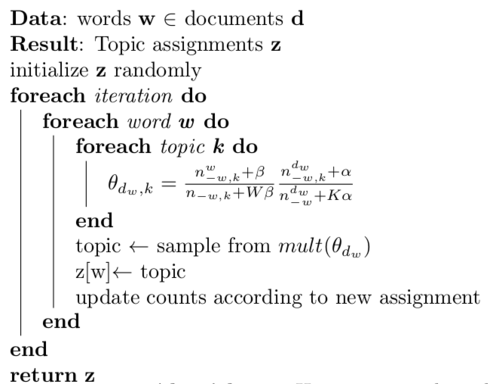
****

Figure 10 LDA Algorithm

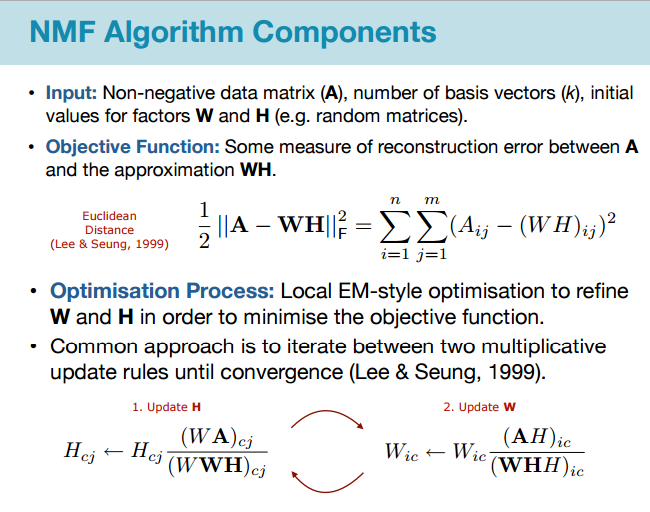


Figure 11 NMF Algorithm

1. **Results Analysis**

We ask the following questions while evaluating the models of NMF and LDA:

**1. How many topics should be learned?**

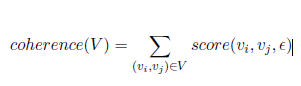
**2. How many learned topics are useful?**

**3. How do these topics relate to often used semantic tests?**

**4. How well do these topics identify similar documents?**

Topic Coherence measures score a single topic by measuring the degree of semantic similarity between high scoring words in the topic. There are two measures mentioned in Mimno et al., 2011; Newman et al., 2010.

Both measures compute the coherence of a topic as the sum of pairwise distributional similarity



Where, *V* is a set of word describing the topic and *e* indicates a smoothing factor which guarantees that *score* returns real numbers

We distill the different models into a shared representation consisting of two sets of learned relations:

* how words interact with topics
* How topics interact with documents.

For a corpus with *D* documents and *V* words, we denote these relations in terms of *T* topics as

(1) A *V × T* matrix, *W*, that indicates the strength

Each word has in each topic, and

(2) A *T × D* matrix, *H*, that indicates the strength

Each topic has in each document.

*T* serves as a common parameter to each model.

|  |  |
| --- | --- |
| **LDA** | **NMF** |
| * Assumes documents are generated by a particular probabilistic model. It first assumes that there are a fixed set of topics, *T* used throughout the corpus, and each topic *z* is associated with a multinomial distribution over the vocabulary, which is drawn from a Dirichlet prior *Dir* (*Beta*). * In this model, the Theta distributions represent the probability of each topic appearing in each document   And the Fi distributions represent the probability of words being used for each topic. These two sets of distributions correspond to our *H* and *W* matrices, respectively.   * Supervised ML Algorithm | * Non-negative Matrix Factorization also factorizes *M* by minimizing the reconstruction error, but   with only one constraint: the decomposed matrices consist of only non-negative values. In this respect,  we can consider it to be learning an un-normalized probability distributions over topics. We use the original Euclidean least squares definition of NMF3.   * Formally, NMF is defined as   *M* = *WH,*where *H* and *W* map directly onto our generalization.   * Semi Supervised ML Algorithm |

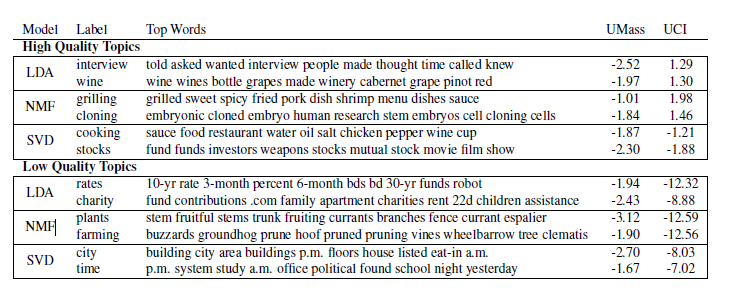
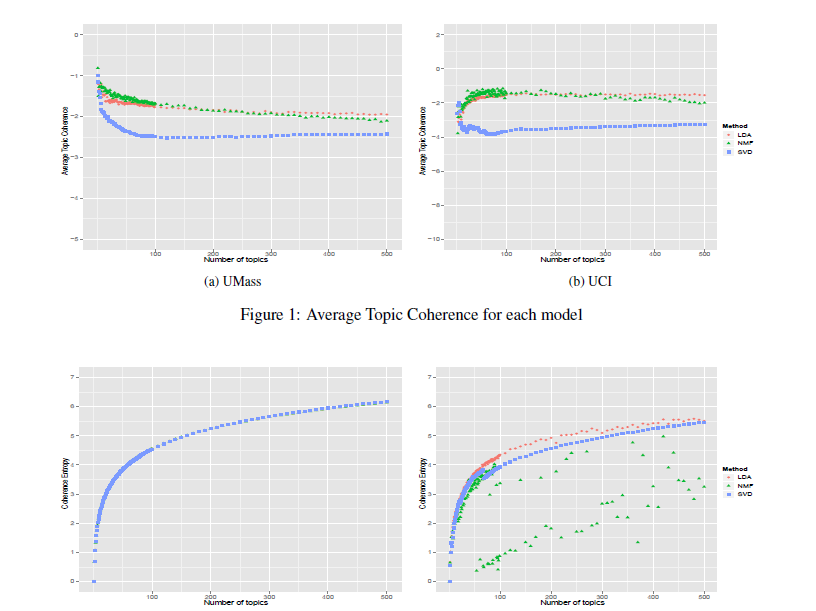


Figure 12 UCI vs UMass Topic Coherence Comparison

The **UCI metric** defines a word pair’s score to be the word probabilities are computed by counting word co-occurrence frequencies.

To some degree, this metric can be thought of as an external comparison to known semantic evaluations.

The **UMass** metric defines the score to be based on document co-occurrence: Significantly, the UMass metric computes these counts over the *original* *corpus* used to train the topic models, rather than an external corpus. This metric is more intrinsic in nature. It attempts to confirm that the models learned data are in the corpus.



1. **Further Enhancements**

This project can be further improved to include many more features:

* We could implement a similar modelling of China’s Research paper dataset and apply different similarity measures to calculate and compare the similarity of research areas in different countries.
* We can predict “implicit” topics by applying centrality measures to the network.

1. **Bibliography**
2. D.D. Lee & H.S. Seung. Learning the parts of objects by non-negative matrix factorization. Nature, 401:788–91, 1999.
3. C. Lin. Projected gradient methods for non-negative matrix factorization. Neural Computation, 19(10):2756–2779, 2007
4. S. Sra & I.S. Dhillon. Generalized nonnegative matrix approximations with bregman divergences. In Proc. Advances in Neural Information Processing Systems (NIPS’05), 2005.
5. Keith Stevens*,*Philip Kegelmeyer,David Andrzejewski,David Buttler,Exploring Topic Coherence over many models and many topics