

# Unsupervised Labelling OF Emails

Vishal Kumawat(10818)

Dibya Ranjan Sahoo(10243)

Advisor: Dr. Amitabh Mukhrjee

Dept. Of Computer Science And Engineering

April 17,2013

## **Introduction:**

Emails have become the basic part of most of our lives. Our communications, information networks etc. are heavily based on emails now a days. We receive a lot of emails per day, as a result the managing these mails becomes quite a daunting task. What we do normally is we classify these emails using the header position of emails like sender, receiver , date . So same receiver , sender ,date come in same folder and we can label these folder. What we want to do is classify these emails according to their semantics . we want to group them according to their content. Mails which have similar kind of information should be in same folder.

In this report we will see how given big data of emails how to classify them according to their content.

## **Related Work:**

Many work have been done related to this problem. Topic Modelling is key of our project. It has lot of application in Text Mining.

Real Time Topic modelling of

Microblogs . the challenge is that in real time we have to extract topics from the stream of blogs which are coming in twitter. The processing of this algorithm is good on large

data. So we can extract topics from this blogs and can tell which topics are people discussing on this application.

Another more similar work is topic modelling on Sarah Palin Emails. In Which there are lot of emails in txt file and we have to get topics from these emails using Topic modelling algorithm.

## Algorithm :

In this project we are using two main algorithms . First one is topic modelling algorithm and another one is K-mean clustering

- **Topic Modelling :** It is probabilistic model Which is based on LDA.

First it randomly assign each word in each document to a topic . so for each document we have a distribution of topics. Now we iterate to improve over topics to find best topic for a given word. Which is done by considering the probability of  $p(\text{topic}|\text{document}) * p(\text{word}|\text{topic})$  .

Topic modelling finally gives us topics . and Each topic have word in it. Which have some kind of hidden pattern. We are free to choose for number of topics. This gives distribution of topic for each document.

- **K-mean Clustering:** It is one of the simplest unsupervised learning algorithm. It classifies data set into k –clusters. The idea for this algorithm is that it defines k centroids and the better choice is place them as far as possible After defining centroid we take each point belonging to given data set. We associate it to nearest point.

## Implementation :

First we want to make a data set of emails in which I have separated the content of emails and header of emails. We applied topic modelling on the content of emails . Topic modelling gives us different topic and words in them. Some topics are more frequent and logical. We choose those topics as features for our k-mean clustering. Number of cluster depend on our data. And we label these cluster according to the

most prominent topic on these cluster.

## Dataset:

We are using 800 emails for our project in which 400 emails are created by us using our personal email box and 400 are taken from online email data set whose link is given below . and we mixed these mails as our new data set.

## Results :

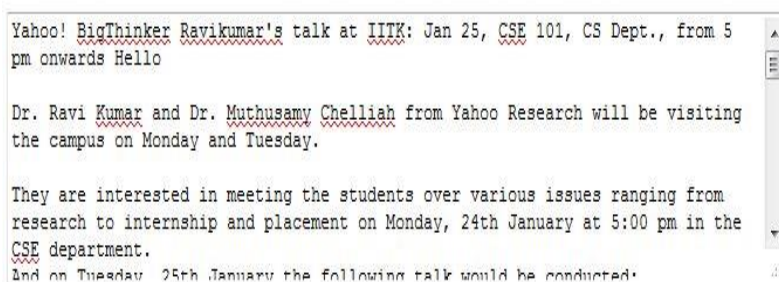
I did topic modelling using 25 topic . because 10 topic is giving many repetition of words in different topics. And high topic no. take more time for topic modelling and k-mean clustering and not much significant improvement in groups. The result of 25 topic is shown below and you can see more detail of result in code section. You can see result of low and high topic number also there.

## Topics

1. gay enron ect randall hou notes rgay dec nsf folders
  2. display edited series ed cambridge springer press science john york
  3. rs lecture details prize venue india iit dr talk organizing
  4. www project world field studies charge book manuscripts cultural ubiquity
  5. events time event head team january techkriti post nominations schedule
  6. digital humanities subject work difference jul texts text making gmail
  7. time due problems questions order automata servants text set work
  8. research project list applications institute subject tei information experience interest
  9. subject reply joyent woodward people question find humanist make point
  10. university http workshop papers conference uk research ac org call
  11. conference de workshops digital proposals workshop subject org university art
  12. department language http english brazil science information literature text natural
  13. time human attention machine dull hope people mind life storytelling
  14. subject david john pm data review james michael procedures original
  15. content cc encoding message bit text type date id transfer
  16. students ac iitk pm april kanpur dear iit hall lists
  17. code programming language software gmail human computer don poetry jun
  18. year form application email day note place program leipzig process
  19. information learning online students education media games hastac www technology
  20. day time gas california power mr people years give cost
  21. pm don week feel good date test subject matthew internet
  22. mail contact rob attached agreement letter call forward vivekananda requested
  23. humanist org digitalhumanities www http php lists listmember humanities interface
  24. enron cn hayslett rod ou na recipients rhaysle privileged pst
  25. http university digital twitter library scholarship humanities www reply electronic
-

Sample of a Document in dat how it distributes on topics:

DOC :137.txt



Yahoo! BigThinker Ravikumar's talk at IITK: Jan 25, CSE 101, CS Dept., from 5 pm onwards Hello

Dr. Ravi Kumar and Dr. Muthusamy Chelliah from Yahoo Research will be visiting the campus on Monday and Tuesday.

They are interested in meeting the students over various issues ranging from research to internship and placement on Monday, 24th January at 5:00 pm in the CSE department.

And on Tuesday, 25th January the following talk would be conducted:

Top topics in this doc (% words in doc assigned to this topic)

- (13%) display edited series ed cambridge springer press science john york ...
- (13%) information learning online students education media games hastac www technology ...
- (12%) rs lecture details prize venue india iit dr talk organizing ...
- (8%) research project list applications institute subject tei information experience interest ...
- (7%) www project world field studies charge book manuscripts cultural ubiquity ...
- (7%) students ac iitk pm april kanpur dear iit hall lists ...
- (7%) events time event head team january techkriti post nominations schedule ...
- (6%) university http workshop papers conference uk research ac org call ...
- (5%) code programming language software gmail human computer don poetry jun ...

Now we have to choose these topics as feature . we can choose all topics as our feature but these does not give us good result . the result of these is given in code section . I am showing here the result of after choosing logical groups as features.

Some major topics used as feature:

Label 2 &3

**iitk/hostel/student related** : [students ac iitk pm april kanpur dear iit hall lists rs lecture details prize venue india iit dr talk organizing](#)

label-4

**project/research related**: [research project list applications institute subject tei information experience interest university http workshop papers conference uk research ac org call](#)

label-6

**coding /programming/software related**: [code programming language software gmail human computer don poetry jun information learning online students education media games hastac www technology](#)

Label-4

**Confrence/workshop/ university related**: [conference de workshops digital proposals workshop subject org university art university http workshop papers conference uk research ac org call](#)

Label-5

**Humanist emails**: [humanist org digitalhumanities www http php lists listmember humanities interface digital humanities subject work difference jul texts text making gmail subject reply joyent woodward people question find humanist make point](#)

Some example of email which have same labels by our clustering:

### Coding/programming/software related

I need to weigh in because, as I suspected, there appears to be a kind of figure-ground problem in the discussion between Elijah and Jim, one we see often in these sorts of discussions.

Jim's point was that *\*code\** cannot be ambiguous or fuzzy to work (with some very minor exceptions around the edges—"standoff markup" and the like, things rarely used in practice though sometimes discussed in theory). I will soften his thesis even more: *programming code* tends strongly toward unambiguous structures and statements, because for the most part it must be interpreted or compiled and then run, and the interpreters and compilers will not accept *\*code\** that is ambiguous.

Elijah appears to be talking about *\*software\** that can function in/handle ambiguous input and actions. I do not believe Jim was doubting that this exists; on the contrary, nearly every *software* program has "emergent properties," "strange behaviors" and so on, and most applications must be able to handle ambiguity of input (to some extent) if it's going to interact with human beings.

Let me, then, reframe Jim's question: the challenge is to provide a significant snippet of *code*, say, a JavaScript function or isolated object from Java or C++ or so on, in which the operating part of the *code* is ambiguous (the compiler could produce multiple correct interpretations) or fuzzy (the compiler can produce no clearly correct interpretation), but the *software* can be compiled and run. Furthermore, because I am interested in tendencies and not so much in absolutes, the challenge is to provide examples of such *code* that are regularly used in everyday applications.

Personally, I do not know of compilers that can actually handle statements that are ambiguous at the level of the program—that is exactly one of the kinds of statements on which a compilers and interpreters are supposed to choke, and it would also violate the definition of the Turing machine out of which all computers are built (for which unambiguous *\*operations\** — not input — are required)—but I am eager to learn.

David

I was specific. I pointed out NetHack, which is a 22-year old game, with freely available source **code** for perusal and is as strange, random and complex as any high gothic novel (You can check it out on Wikipedia, though the synopsis will only hint at the strangeness and subtlety of the gameplay). **Games** are analogous to fiction in writing, whereas operating systems, spreadsheets and metadata collation **software** is analogous to technical writing. So if you're looking for interesting coding, you should look to the right genre. I'm sure there are some great turns of phrase to be found in the corpus of lawnmower assembly manuals, but I don't think they're a good indication of the state of Western literature. Granted, most modern, big budget games are as interesting as big budget movies and books, but there's a real wealth of quirky, strangely programmed and functioning **games** out there. The art of writing a game, most especially the older and smaller **games**, with their connection to random numbers to represent chance, is clearly similar to the creation of prose and poetry. There are entire sections of **code** in some of these **games** that never get performed except under the most esoteric of circumstances, and there are interesting emergent properties of the interacting game world that capture the imagination of players and coders, regardless of user input. And yes, you can dip your feet into it with only a knowledge of XML or Perl. The game modification community has grown so large and the modification of **games** has grown so pervasive that many companies create specific entry points into modifying game content through creation of XML files or writing simple scripts.

I'm not sure how you mean the question, "why would anyone use this except for personal projects?" though. What is the "use" of poetry or literature? How is a collection of the poetry of Emily Dickinson more useful than the aforementioned lawnmower manual? It's drudge work writing a lawnmower manual, or an academic paper, but we don't claim that therefore people shouldn't learn to write. If, however, one feels that the writing of literature is of value and its structures should be analyzed (and therefore understood to some meaningful extent) then it would seem the same would apply to creative **software** and there would be an incumbent need to be literate enough to analyze and understand it. You wouldn't blame a **software** engineer for not liking poetry, but you'd likely think him an idiot if he claimed poetry did not have the ability to pass along complex truths in the way that **software** does and therefore that he didn't need to learn how to read.

This is a rather long and scattered response, but it's no longer clear to me your exact criticism. Is it that you think that low-level **programming** languages don't allow for the creation of nuanced, complex thought, or is it that you feel that **code** is goal-oriented and utilitarian or is it simply that **software** is inherently boring? I believe I've addressed the first two and, as for the third, I think that the all-pervasive nature of **software** militates against treating it as an ignorable subject. We have a basic expectation of literacy due to the pervasive nature of writing, and I think that we should have an equal expectation of **software** literacy. So, whereas Dr. McCarty's originally framed the question of **software** literacy (A term I've used without defining, but which I assume involves a working knowledge of creating **software**) in terms of fear, I feel it's more related to underestimating the scope and value of **software** as metaphor, creative work and tool.

## NEW JOURNAL COVERS HIGHER ED INFORMATION LITERACY

The NORDIC JOURNAL OF INFORMATION LITERACY IN HIGHER EDUCATION, published by the University of Bergen, is a peer-reviewed, open-access journal created to encourage "research-based development of information literacy teaching within the educational programmes of universities and higher education colleges" and to establish "a forum for the investigation and discussion of connections between information literacy and general **learning** processes within subject-specific contexts."

Papers in the inaugural issue include:

"A New Conception of Information Literacy for the Digital Environment in Higher Education" by Sharon Markless

To provide an information literacy (IL) framework for a virtual **learning** environment, the author considered the "relevant principles of **learning**, the place of student reflection when **learning** to be information literate, what IL in higher education (HE) should encompass, the importance of context in developing IL, and the influence of the digital environment, especially Web 2.0."

"Google Scholar compared to Web of Science. A Literature Review" by Susanne Mikki



## **Improvements and Conclusion:**

We can use the header information like subject and whether the email was sent to single or multiple receiver. In our project we have classified the emails based on balance between email header information and email semantics information. We have also found a parameter which give priority to email header or email semantics.

## **References:**

Ozcaglar, Cagri(2008)

Topic Modelling theory-

<http://clc.yale.edu/2011/10/07/how-to-do-your-own-topic-modeling/>

<http://www.fredgibbs.net/cliow3workspace/blog/topic-modeling/>

<http://miriamposner.com/blog/?p=1335>

<http://blog.echen.me/2011/06/27/topic-modeling-the-sarah-palin-emails/>

Topic Modelling Tool-

<http://nlp.stanford.edu/software/tmt/tmt-0.4/>

Dataset-

<http://dhumanist.org/Archives/Current/>