Enron: E-mail Body Analysis using Topic Modeling

Sagar Singh
MSDS
Northeastern University
Boston, MA
singh.sag@husky.neu.edu

Sinjini Bose
MSDS
Northeastern University
Boston, MA
bose.s@husky.neu.edu

INTRODUCTION

In the year 2000, Enron was one of the largest energy companies in America. However, after being outed for fraud, it spiralled downwards into bankruptcy within a year making it one of the largest corporate meltdowns in history. After it's downfall, Federal Energy Regulatory Commission made their data (consisting of 0.5 Million messages exchanged across 150 senior officials) public to aid in ongoing investigations. The goal of our project is to apply Topic Modeling to categorize emails into topics and highlight most commonly discussed topics among employees.

By categorizing emails into set of topics, we might be able to detect some anomalous patterns (since the company was called out for fraudulent activities) within the emails, that might have led to further disruptions within the company. While our analysis, would be very specific to the Enron corpus, but this could further be generalised for other organisations, for early detection of anomalous mail exchanges (if any) within employees. This can also help in mailbox usage optimization technique such as flagging of redundant emails.

RELATED WORK

While some of the researches have already been conducted on Enron dataset, but none in the line of topic modelling.

1. **Crowdsourcing Evaluations of Classifier Interpretability**

   This was a supervised machine learning based project, aimed towards comparing human vs computer generated classifications. Enron email dataset was used as part of test corpus and labels were assigned based on whether an email is work related or non-work related. Members were then assigned a set of email and were asked to determine its class and highlight words that supported this classification.

   The results were then compared to check if there was any significant distinction between how humans preferred labels and how the computer classified those emails. For Enron, it was found that computer-generated explanations and human-generated explanations were not always distinguishable.

2. **Social Network Analysis**

   This project was based on network analysis, to trace links between the employees using count of emails exchanged between them. This project was initially focused on identifying links for Bill Williams, one of the main suspect who was directly involved with manipulation of energy production. They then moved forward to identify the links among the top employees at Enron and look for similar patterns like that of Bill Williams.

   Through this analysis they were able to highlight immediate connections among the top employees and were successful in identifying hidden management structure within Enron.

BACKGROUND INFROMATION

Based on the data released by Federal Energy Regulatory Commission in 2003, it was discovered that top employees belonging to the ranks of CEO and senior executives of departments such as Treasury, Internet Unit and Legal Department, were indeed involved in fraudulent activities. With an attempt to analyze the volume of mails exchanged between certain employees, few of the previous researches were able to trace back the links to former Directors and CEO of the company.

The whole act of committing fraud started around 1996 with the then CFO Andrew Fastow. He created Chewco in an effort to hide debt and inflate profits. During 1998, Enron took part in several capital-intensive ventures and after a series of financial disaster, it was then decided by the board of directors to run private companies that do business with Enron. One such company was LJM that used to buy poorly performing Enron assets but in reality, was used to hide debts and inflate stock prices.

The company officials, during later years (2000-01) began filing fraudulent quarterly reports. On one hand where losses were piling up, some of the senior members (like Jeffrey Skilling and Robert Belfer) started selling major chunks of their stocks further leading to disruptions within the company. After a series of losses, Enron filed for bankruptcy around early December and it was then speculated that the company might have violated several securities law.

PROPOSED APPROACH

The problem at hand, is to figure out whether the energy company's financial distress, resulting from fraudulent activities was discussed among its employees through mail exchanges. The analysis is therefore, directed towards extracting the key topics of the email threads, that would capture the essence of the discussions.

From the large collection of emails, consisting of the attributes – 'To', 'From', 'Date', 'Message_body', 'Subject', 'Folder_name', 'Cc', the ones that were selected for the analysis were –

1. The field 'Message_body' which is the body of text representing the email threads
2. The fields 'To' and 'From', which were mainly used in the analysis for filtering out duplicate entries of mails.
3. The field 'Subject' which was used for identifying mails that were forwarded.

In order to capture the essence of the discussions, the analysis implements topic modeling using LDA (Latent Dirichlet Allocation) on the email bodies. Since the analysis is carried out in Python, we used the implementation of LDA that is available in Python's Gensim Library **(gensim.models.ldamodel.LdaModel**), with Variational Bayes Sampling for distribution of topics. LDA is used for discovering the Latent or hidden factors/topics that influence a particular body of text and since such complex distributions can't be sampled directly, Gensim uses Variational Bayes sampling, as opposed to Gibb's Sampling and Monte Carlo sampling methods for better speed optimization. It also supports multicore approach for model building, which in turn makes the process of fitting the data to the model faster.

Variational Bayes can be seen as an extension of the EM (expectation maximization) algorithm from maximum aPriori estimation (MAP estimation) of the single most probable value of each parameter to fully Bayesian estimation which computes (an approximation to) the entire posterior distribution of the parameters and latent variables. As in EM, it finds a set of optimal parameter values, and it has the same alternating structure as does EM, based on a set of interlocked (mutually dependent) equations that cannot be solved analytically.

Why choose LDA for the problem at hand?

LDA is a statistical generative model, that would take into consideration the latent variables that indicate how a particular document is a mixture of topics/keywords. In other words, it assigns probabilities or weights for a particular document to belong to a collection of topics, with some topics being dominant.

With emails, we can never be sure, if all emails talk about just one category of discussion, or whether they involve discussions related to multiple topics, hence other text analysis methods such as key word extraction or key phrase extraction, might not be really helpful in representing what a mail is all about, and so we find LDA to our rescue. In this approach, each email body has been chosen as a document and the topics which we get from LDA are the words or tokens that are present in the mail bodies.

EXPERIMENTAL SETUP

The e-mail exchanges between 150 Enron employees, arranged in folders (where one folder belonged to each employee), available in text format is used as the primary source of data for this analysis.

Data Preparation:

1. The text files were collated and converted to a single .csv file, where every row represented a single email, exchanged between the employees. The Columns of the .csv are the attributes of the e-mails (as discussed above in the 'Proposed Approach') – only the important attributes were considered further in the analysis.

2. The body of text present in the employee folders, didn't have the fields or the attributes categorized – to put it simply, it was just a paragraph containing the attributes 'From', 'To', 'Subject', 'Message Body' , 'CC', 'Date' , all in one place. Python's library 'Email' was used in separating the attributes from the text body, so that they could be stored in different columns of a Pandas DataFrame.

3. After retrieving the necessary information and storing it in a DataFrame, the Analysis moves ahead to find if there are any duplicates present in the data, or in other words, if complete rows within the DataFrame have been duplicated. Finding duplicates and removing them from the final data is important because they can give rise to unnecessary biases, in the final part of the analysis.

   For example: if a certain mail is repeated a couple of times, then after it can be considered to have a higher weightage over the other emails, when actually it isn't, it's count is just higher as a result of duplicity. The reason for having duplicates in the data, is that at times employees stored the same email in two different folders, and when the data from every folder was extracted, the mails got repeated.

   However, in this effort to remove the duplicates, the mails that turned out to be redundant, because they were forwarded, were not removed from the finally data, because unlike the other redundant emails these actually held some importance (and hence were forwarded), and hence it is only logical that they should be provided with a higher weightage while finding the topics. The Forwarded mails were identified by 'Fw:' tag present in the subject line of each email (if forwarded).

4. **Number of rows:**

Total messages in original dataset	0.5 Million
Count of duplicate emails	257263

# of forwarded emails present in duplicate emails	11250
Final Count of rows	0.5M – 257263 + 11250 ~ **242099**

5. Since in this particular analysis we are interested only in mails that were exchanged within the organization Enron, the Senders and Receivers for each mails were inspected, using the attributes '*From'* and '*To'*, and all such mails that were sent from an external id (where external id is any id that doesn't have the address '@enron.com' attached to it), to another external id, were purged. However, emails that were sent from external senders to Enron employees, are considered in the analysis.

Data Preprocessing:

6. The final data, consisting only of the email bodies (after removing all the duplicates and unrequired mails) was then tokenized – the **sentences were split into words or tokens**. Python's NLTK library provides a word tokenizer, which has been used for this particular purpose.

7. Some of the **unwanted punctuations were then removed** which helped in analyzing the texts better and allowed to get a gist of the data.

8. For certain significant sample of the emails, it was found that they were heavy on phone numbers, email IDs and other website links, which would just result in appearance of such entities in the final topics. So, the **phone numbers were replaced by a single word 'number' and the emails and links were represented by a word 'email/link'**, which would make sure that if any topic comes up with a term 'number' or 'email/link' then it should be indicative that, the email where that particular topic is dominant, is heavy on such entities.

9. It was also identified that at times, while writing the emails, employees missed out on whitespaces between two sentences, and started a new sentence, after a period. Since, the word tokenizer used in the analysis considers space as a delimiter, it wasn't able to identify two different words that didn't have a space between them and treated them as a single entity. So, we **applied regex, to separate such words into two separate tokens**.

10. In the next step, **POS (Parts of Speech) tagging** and parsing of each token was done, and the tokens that fell into the category of Proper Nouns, Cardinal numbers, Modal verbs and Prepositions were removed from the set of tokens. This is because, Proper nouns don't significantly add to the value of any topic (**Example**: someone's name or the name of a month will just increase noise in the topics). Similarly, prepositions, modal verbs and cardinal numbers are just filler words that we use while writing sentences and hence don't contribute significantly towards the formation of any topic.

11. Next, **Stopwords filtering and Lemmatization** was performed on the set of tokens. Stopwords are a list of predefined tokens/words that are used in the colloquial language, again these can also be termed as filler words, and hence don't contribute significantly to any kind of text analyses. Lemmatization is process of removing the suffix of a word, while preserving its root form – basically the POS of the particular word is first checked and then if it is not present in its root form already, the Lemmatizer converts it, based on its POS. The analysis uses, the Wordnet Lemmatizer and Stopwords set as available in Python's NLTK library for the above-mentioned purposes.

12. The final refined tokens are then **converted to a dictionary**, where each word is assigned an ID. A corpus is formed out of all the available unique tokens along with their frequencies, or the number of times that the particular token appeared in a document (in this case an email). The corpus and dictionary are then passed as arguments to the LDA model.

Model Training:

13. This particular implementation of LDA requires **number of topics as one of the parameters**. The heuristic applied to choose the number of topics in this analysis is the evaluation of Coherence scores.

    Gensim provides four options for coherence score and out of those, we have used '**c_v**' coherence score, which is based on sliding window approach, for determining appropriate number of topics. The counts are used to calculate the NPMI (Normalized Point Wise Mutual Information) of every top word to every other top word, thus, resulting in a set of vectors—one for every top word.

    The one-set segmentation of the top words leads to the calculation of the similarity (based on cosine) between every top word vector and the sum of all top word vectors. The coherence is the arithmetic mean of these similarities.

14. **Coherence Scores** for different values of topic number:

Coherence Score for 10 topics	0.4007
Coherence Score for 15 topics	0.3213
Coherence Score for 20 topics	0.3225

The number of topics giving the highest coherence score is chosen, to get the final results (here 10 topics).
Although mathematically, it is proven that the number of topics having the highest coherence score should be chosen, qualitatively any number of topics, that results in human interpretable topics, should be chosen i.e., there should be a fair trade-off between the mathematical and the qualitative results.

To evaluate the results, we're also taking into account **Perplexity Score** (the lower the perplexity scores the better)

Perplexity score for 10 topics	-9.6686
Perplexity score for 15 topics	-14.0514
Perplexity score for 20 topics	-20.2382

Based on the above statistics and qualitative analysis of the topics generated, we chose number of topics as 10 for our final model

15. Gensim also provides the option for batch processing, known as online iterative learning, and in this analysis the online iterative learning is used, as it is relatively faster with no significant negative effect on the results. This option can be chosen by setting the parameter '*update_every*'=1 in the **gensim.models.ldamodel.LdaModel** implementation.

Further Exploration:

16. **Bigrams** are used in the next step of the analysis, to check if significantly better topics could be generated, than the topics obtained from using single tokens.

Coherence Scores for bigrams:

Coherence Score for 10 topics	0.6465
Coherence Score for 15 topics	0.6506
Coherence Score for 20 topics	0.5865

As evident form the above scores, there is a significant rise in the coherence scores for bigrams, when compared with previous scores. Also, on analysing the topics qualitatively, the terms formed (using bigrams), in each topic tend to make more sense and are better in terms of interpretation.

Perplexity Scores for bigrams:

Perplexity score for 10 topics	-18.1970
Perplexity score for 15 topics	-23.5890
Perplexity score for 20 topics	-32.8524

RESULT AND DISCUSSIONS

For **Single Tokens** with **number of topics as 10** we obtain the below terms for each of the significant Topics

- **For Topic 1(34.9% of tokens)**

  Top Terms associated with this topic are: '0.019*"meeting" + 0.018*"employee" + 0.013*"also" + 0.012*"help" + 0.012*"year" + 0.011*"work" + 0.010*"want" + 0.010*"make" + 0.009*"people" + 0.009*"last"'

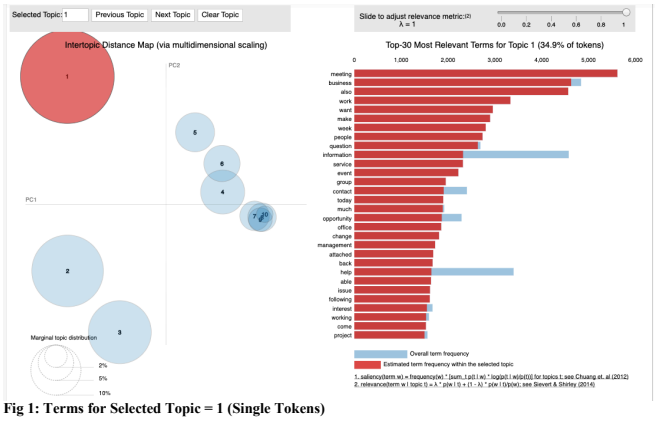

Fig 1: Terms for Selected Topic = 1 (Single Tokens)

- **For Topic 2(20.4% of tokens)**

  Top Terms associated with this topic are: '0.061*"company" + 0.026*"energy" + 0.025*"market" + 0.017*"year" + 0.017*"stock" + 0.013*"consumer" + 0.011*"price" + 0.011*"power" + 0.009*"state" + 0.008*"country"'

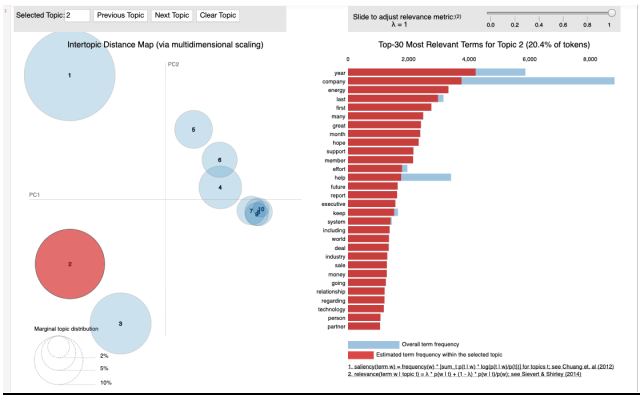

Fig 2: Terms for Selected Topic = 2 (Single Tokens)

- **For Topic 3(15.8% of tokens)**

  Top Terms associated with this topic are: '0.061*"company" + 0.026*"energy" + 0.025*"market" + 0.017*"year" + 0.017*"stock" + 0.013*"consumer" + 0.011*"price" + 0.011*"power" + 0.009*"state" + 0.008*"country"'

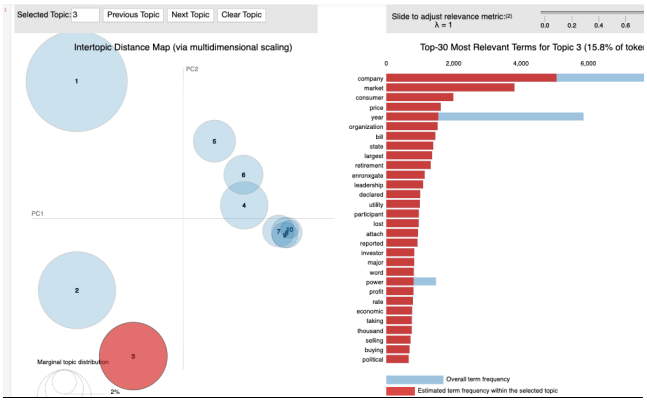

Fig 3: Terms for Selected Topic = 3 (Single Tokens)

Top 10 Terms for each Topic using Single Tokens

```
Out[356]:
[(0,
  '0.089*"conference" + 0.038*"contribution" + 0.025*"relief" + 0.022*"date" +
  0.022*"number" + 0.021*"budget" + 0.016*"reminder" + 0.015*"committee" +
  0.013*"appreciated" + 0.012*"get"'),
 (1,
  '0.061*"company" + 0.026*"energy" + 0.025*"market" + 0.017*"year" + 0.017*"stock" +
  0.013*"consumer" + 0.011*"price" + 0.011*"power" + 0.009*"state" + 0.008*"country"'),
 (2,
  '0.015*"said" + 0.012*"interest" + 0.011*"project" + 0.010*"contract" + 0.009*"deal" +
  0.009*"industry" + 0.009*"sale" + 0.009*"going" + 0.009*"including" + 0.008*"offer"'),
 (3,
  '0.049*"feedback" + 0.040*"home" + 0.026*"final" + 0.022*"heart" + 0.019*"brief" +
  0.018*"desktop" + 0.018*"client" + 0.015*"period" + 0.015*"transacted" +
  0.015*"transacting"'),
 (4,
  '0.166*"enronxgate" + 0.047*"ipaq" + 0.012*"ensuring" + 0.012*"fraud" + 0.008*"flip" +
  0.005*"enronxgateto" + 0.004*"draft" + 0.003*"inadvertantly" + 0.003*"extend" +
  0.000*"ti=me"'),
 (5,
  '0.070*"business" + 0.035*"service" + 0.020*"system" + 0.018*"technology" + 0.016*"cost"
  + 0.015*"success" + 0.014*"million" + 0.014*"medium" + 0.013*"candidate" +
  0.010*"server"'),
 (6,
  '0.019*"meeting" + 0.018*"employee" + 0.013*"also" + 0.012*"help" + 0.012*"year" +
  0.011*"work" + 0.010*"want" + 0.010*"make" + 0.009*"people" + 0.009*"last"'),
 (7,
  '0.047*"information" + 0.028*"question" + 0.024*"contact" + 0.018*"send" +
  0.017*"report" + 0.016*"request" + 0.016*"attached" + 0.016*"available" + 0.015*"list" +
  0.015*"following"'),
 (8,
  '0.055*"received" + 0.033*"leadership" + 0.031*"writing" + 0.026*"little" +
  0.023*"yesterday" + 0.023*"afternoon" + 0.022*"training" + 0.014*"student" +
  0.013*"happen" + 0.012*"faith"'),
 (9,
  '0.062*"case" + 0.056*"participation" + 0.054*"http" + 0.024*"active" + 0.024*"ticket" +
  0.021*"live" + 0.019*"internet" + 0.019*"busy" + 0.018*"internal" + 0.018*"club"')]
```

Fig 4: Top Terms for each Topic Type (Single Tokens)

For **Bigrams** with **number of topics as 10** we obtain the below terms for each of the significant Topics.

- **For Topic 1(90.5% of tokens)**

  Top Terms associated with this topic are: '0.007*"company declared" + 0.005*"energy bill" + 0.005*"enronxgate enronxgate" + 0.004*"many employee" + 0.004*"million dollar" + 0.004*"summaryexternal transacted" + 0.004*"energy crisis" + 0.004*"writing donate" + 0.004*"reported sold" + 0.004*"bankruptcy fund"'

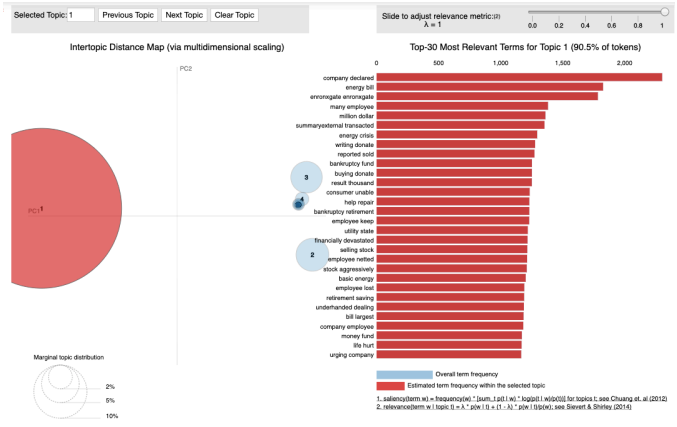

Fig 5: Terms for Selected Topic = 1 (Bigrams)

- **For Topic 2(3.8% of tokens)**

  Top Terms associated with this topic are: '0.005*"intended recipient" + 0.004*"last night" + 0.003*"information contained" + 0.003*"contain confidential" + 0.003*"recipient contact" + 0.003*"meeting conference" + 0.002*"prohibited intended" + 0.002*"theintended recipient" + 0.002*"join congratulating" + 0.002*"sport event"'

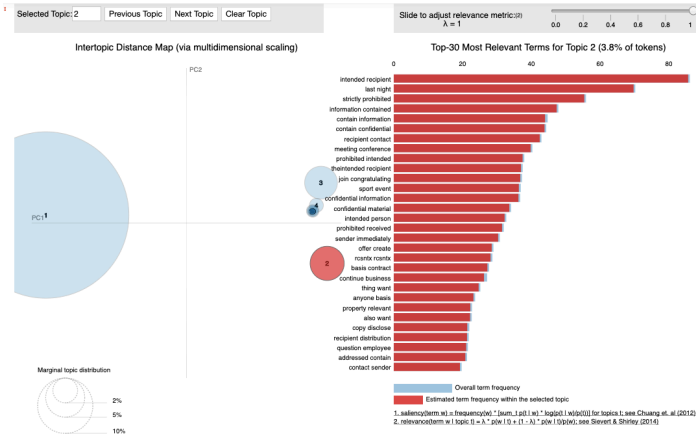

Fig 6: Terms for Selected Topic = 2 (Bigram)

Top 10 Terms for each Topic using Bigrams

```
Out[354]:
[(0,
  '0.006*"transaction stock" + 0.006*"business unit" + 0.004*"report question" + 0.004*"peer group" +
  0.003*"come back" + 0.003*"required report" + 0.003*"stock option" + 0.002*"meaningful feedback" +
  0.002*"including purchase" + 0.002*"cell phone"'),
 (1,
  '0.007*"company declared" + 0.005*"energy bill" + 0.005*"enronxgate enronxgate" + 0.004*"many
  employee" + 0.004*"million dollar" + 0.004*"summaryexternal transacted" + 0.004*"energy crisis" +
  0.004*"writing donate" + 0.004*"reported sold" + 0.004*"bankruptcy fund"'),
 (2,
  '0.000*"based date" + 0.000*"city hotel" + 0.000*"substantial personal" + 0.000*"sure reasonable" +
  0.000*"system always" + 0.000*"system remains" + 0.000*"threatens individual" +
  0.000*"unsatisfactory return" + 0.000*"another site" + 0.000*"availability departure"'),
 (3,
  '0.003*"soon possible" + 0.001*"price volatility" + 0.001*"customer deal" + 0.001*"short position" +
  0.001*"send copy" + 0.001*"week still" + 0.001*"release send" + 0.000*"logistical issue" +
  0.000*"sure everything" + 0.000*"letter board"'),
 (4,
  '0.026*"last year" + 0.005*"question contact" + 0.004*"requested feedback" + 0.004*"keep informed" +
  0.003*"former employee" + 0.003*"hard copy" + 0.003*"management style" + 0.003*"assessment
  assessment" + 0.003*"high level" + 0.003*"complete request"'),
 (5,
  '0.005*"intended recipient" + 0.004*"last night" + 0.003*"information contained" + 0.003*"contain
  confidential" + 0.003*"recipient contact" + 0.003*"meeting conference" + 0.002*"prohibited intended"
  + 0.002*"theintended recipient" + 0.002*"join congratulating" + 0.002*"sport event"'),
 (6,
  '0.002*"question comment" + 0.001*"wholesale power" + 0.000*"attached stock" + 0.000*"local
  distribution" + 0.000*"regarding consent" + 0.000*"nothesitate contact" + 0.000*"second inthe" +
  0.000*"schedule approval" + 0.000*"request forwarding" + 0.000*"relates betweeni2"'),
 (7,
  '0.001*"last couple" + 0.001*"regular basis" + 0.000*"intellectual property" + 0.000*"rate basis" +
  0.000*"staying late" + 0.000*"tax deferred" + 0.000*"spoke loan" + 0.000*"specifically also" +
  0.000*"trip planning" + 0.000*"trip tomorrow"'),
 (8,
  '0.001*"meeting held" + 0.000*"conflict contact" + 0.000*"tomorrow regard" + 0.000*"meal =the=20" +
  0.000*"l=et=20me want" + 0.000*"healthy hearty=018" + 0.000*"idea =20dilemma" + 0.000*"in==20the
  normal" + 0.000*"door knock" + 0.000*"skimmel iexalt"'),
 (9,
  '0.004*"strictly prohibited" + 0.000*"received thiscommunication" + 0.000*"service lawsuiton" +
  0.000*"serve inappropriatetime" + 0.000*"sort action" + 0.000*"shareholder act" + 0.000*"suit class"
  + 0.000*"sandy_tungare defendantin" + 0.000*"yourbehalf document" + 0.000*"recovered lawsuit"')]
```

Fig 7: Top Terms for each Topic (Bigrams)

CONCLUSIONS AND FUTURE WORK SECTION

The objective of this analysis was to indicate about any conversations about the company Enron's on-going financial troubles and fraudulent activities, amongst its employees. The topics that were generated, from topic modeling of both the single tokens, and bigrams were indicative of topics such as - energy and power stock prices (since Enron was an energy-based company), financial shortcomings, investigations related to frauds- which meets our objective. The topics generated from LDA model using bigram tokens were better in terms of interpretation, than the ones generated from the LDA model using single tokens, which was indicated both by their coherence scores, and also by a simple glance at the topics.

Percentage Contribution of Dominant Topics with mail body:

Percentage Contribution of Dominant Topics to a mail body With	% Contribution	Topics Keyword	Email message
Single Tokens	0.5603	meeting, employee, also, help, year, work, want, make, people, last	Susan Kuntz276 Hamilton St.Rochester, NY 14620kuntz@geneseo.eduTo Mr. Ken Lay,I'm writing to urge you to donate the millions of dollars youmade from selling Enron stock before the companydeclared bankruptcy to funds, such as Enron EmployeeTransition Fund and REACH, that benefit the company'semployees, who lost their retirement savings, and providerelief to low-income consumers in California, who can'tafford to pay their energy bills. Enron and you mademillions out of the pocketbooks of California consumersand from the efforts of your employees.Indeed, while you netted well over a $100 million, many ofEnron's employees were financially devastated when thecompany declared bankruptcy and their retirement planswere wiped out. And Enron made an astronomical profitduring the California energy crisis last year. As a result,there are thousands of consumers who are unable to paytheir basic energy bills and the largest utility in the state isbankrupt.The New York Times reported that you sold $101 millionworth of Enron stock while aggressively urging thecompany's employees to keep buying it. Please donate thismoney to the funds set up to help repair the lives of thoseAmericans hurt by Enron's underhanded dealings.Sincerely,Susan Kuntz
Bigrams	0.8946	company declared, energy bill, enronxgate enronxgate, many employee, million dollar, summaryexternal transacted, energy crisis, writing donate, reported sold, bankruptcy fund	Enron has consented to allow the Federal Bureau of Investigation to search the Enron Building and to interview employees regarding allegations of document destruction. The company encourages all employees to fully cooperate with the FBI.The company has retained Michael Levy and his law firm Swidler Berlin, Shereff, Friedman, to advise any employee regarding this matter and to be present with them during any interviews by government investigators. If you would like to speak with an attorney, please contact Ned Crady of the Legal Department at 3-4558 or contact Michael Levy's office in Washington, D.C. at 202-295-8414 and they will direct you to him here in Houston.

Fig 8: Using Single Tokens/Bigrams we are able to capture the essence of what's being spoken around

Although our results are pretty indicative of the ongoing activities at Enron, these results were obtained by fitting the model on a subset of the data, due to memory limitations of the machine. In future, if we carry out the analysis and build the model on the whole dataset, then there are chances that the analysis could generate results, which are more insightful.

REFERENCES

[1] Dataset: https://www.cs.cmu.edu/~./enron/

[2] Crowdsourcing Evaluations of Classifier Interpretability: https://www.aaai.org/ocs/index.php/SSS/SSS12/paper/viewFile/4267/4689

[3] Using Social Network Analysis Measures: https://cambridge-intelligence.com/using-social-network-analysis-measures/

[4] Gensim LDA Package: https://radimrehurek.com/gensim/models/ldamodel.html

[5] Gensim White Paper: https://radimrehurek.com/phd_rehurek.pdf

[6] Enron Investigation: https://www.google.com/search?client=safari&rls=en&q=enron+scam+ppt&ie=UTF-8&oe=UTF-8#