Introduction to Machine Learning Project Report

Monsoon 2019

Project GitHub Link

Archit Checker, Deepraj Pandey, Niyati Bafna

December 1, 2019

Abstract

The Enron email dataset Kaelbling and Gervasio (2015) is a corpus of 517401 raw emails, of the employees of the Enron Corporation. It was released publicly after the Enron scandal i.e. siphoning of money by high-level executives, in the hope of identifying the perpetrators of the fraud. We attempted to cluster the emails based on their content and try to create clusters of emails with similar topics of discussion. This is a meaningful intermediate step to the main problem because it potentially narrows down the search into emails discussing relevant topics, such as funds, project dealings, and accounting, to find the senders of problematic emails. We tried three main approaches: K-Means clustering on TF-IDF values, Latent Semantic Indexing, and Latent Dirichlet Allocation, and concluded that topic modelling using LDA was the most appropriate approach, due to evaluation by several metrics, such as Hopkins Statistic, silhouette scores, topic coherence (c_v and u_mass), and perplexity. Finally, we ran LDA over a range of topic numbers, and gained the best results for segregation into 5 topics. This model gave us a coherence score 0.582.

Related Work

An interesting prior approach for intent classification using the enron dataset is discussed in Sappelli et al. (2016). This approach aims to eliminate information overload and finds that a model trained over the corpus of one corporate can be used to accurately classify E-mails for another corporation as well. However, this approach relies on manually annotating E-mails for intent and hence is a supervised approach, depending on annotating resources.

The approach followed in Kumar and Aldous (2015) is similar to ours, but uses LDA only with cluster size 4 without further evaluation, and without comparison with other approaches or models. There is scope for a more detailed analysis here and we treated this approach as a starting point for our project.

During our literature review, we also came across Repke et al. (2018) which uses Latent Dirichlet Allocation for topic modelling but since it is a part of a larger software toolkit, we do not see an analysis of the approach in the paper. Further readings on LDA were aided by Blei et al. (2003) and Hermans and Murphy-Hill (2015).

Approach

Initial Approaches

We had some idea of classifying documents into themes based on a manually calibrated set of themes. To do this (hypothetically), we would first extract all the nouns and verbs over all documents to create a complete clean and lemmatized vocabulary. We made the assumption that 'Themes' are contained in only these two Part-Of-Speech categories.

We would then manually slot them into themes. Now, given a new document, we could think of it as a bag of words, and

1. Over all the words in the document, find the maximum occurring theme.
2. Improve that measure by weighting each term by its TF-IDF value.

This is not a good approach, because

1. We are ignoring the fact that a word may belong to two or more themes when we calibrate it into one theme.
2. We are not accounting for the fact that different words may indicate a certain theme to different extents e.g. ‘tennis’ generates ‘racquet’ with a probability of 0.1 but ‘ball’ with a probability of only 0.01. Therefore, we cannot do a simple count of ‘tennis’ words. (Note that TF-IDF also does not cover this failure, as it will only weigh words according to their own frequency in documents, but not in themes.)
3. We are not taking the help of statistics and relying instead on human judgment. For example, a human may not be very sure that ‘football’ is an indicator of ‘college’ over ‘job’, but a machine may infer that statistically, football is used more in discourse about college, and so is generated more from that theme.
4. Manual calibration is a highly noisy process - it is extremely subjective and will vary from human to human. This is not even to mention the subjectivity of words themselves.

Of course, we did not go ahead with this idea. Instead, we thought that since the problem seemed to be mostly with human inadequacy in theme-categorization and introduction of our own biases, we could try to use an unsupervised algorithm to perform clustering – that is, the K-Means clustering algorithm.

K-Means

This was the second possibility. We thought we could represent the documents as TF-IDF and cluster on that basis. We could then later view the documents that had been clustered together, and attempt to calibrate the clusters.

Before we went ahead with the clustering, we decided to get an idea of how meaningful it was to run K-Means on this data and find out how clusterable it was.

Clustering Tendency

To figure out the clustering tendency, we used two approaches. The first of these was the Hopkin’s Statistic.

The Hopkin’s Statistic works in the following way for a Dataset D of size n-

1. Null hypothesis: The data is uniformly generated (it is not amenable to clustering).
2. Choose a sample size: m. Choose a random sample X of m points x_i from the dataset.
3. Generate Y, a set of m points y_i from a uniform random distribution.
4. Define u_i: distance of y_i from nearest neighbour in D
5. Define w_i: distance of x_i from nearest neighbour in D

Then,

$$H = \frac{\sum_{i=1}^{m} u_i^d}{\sum_{i=1}^{m} u_i^d + \sum_{i=1}^{m} w_i^d}$$

For very small samples, we got unreasonably high values. For instance, for 500 emails we got a value of 0.898.

Unfortunately, we could not run the implementation of this on the entire dataset due to logistical issues. Following this, we switched to evaluating clusterability using the silhouette score.

The basic idea of the silhouette score is to calculate a ratio of the intra-cluster sum of squares and the inter-cluster sum of squares. It is a measure between -1 to $+1$.

On calculating this score, we got the following results

1. 7 clusters : $0.009, 49448.217$
2. 11 clusters: $0.01, 48986.159$

These are low values of silhouette score, and high values for inertia (although it is difficult to say what is high with inertia since it varies from dataset to dataset). This suggests that our dataset is not very well suited for clustering in this way.

Concluding remarks on K-means

We took a look at the files in the 7 different clusters (first case). To get an idea of the 'theme' of the cluster if any, we ran another TF-IDF on the seven document clusters, treating each as one large document to find terms that most 'characterized' each cluster. The clusters we obtained from K-means can be observed here - https://checker5965.github.io/kmeans.txt

Notice 'variance', 'data', 'error', 'detect', 'detail' in cluster 1, and perhaps 'agreement', 'review', 'draft' in cluster 5. However, as expected, we do not see the topics or the corresponding emails holding together very much in a significant way on the whole.

At this point, it was not clear to us whether clustering based only on TF-IDF would mean anything. A possible conceptual flaw to this approach was that this clustering might not give us a topic-based classification at all. Since TF-IDF is quite a loose representation of a document, based only on word and document frequencies, we cannot be sure that the model has actually learned topics and not something else, if at all.

Therefore, we thought that a better representation of the documents might give us more meaningful clusters. After a little more research, we decided to pick topic modelling as our second approach because that seemed like a more direct solution to our problem statement.

Topic Modelling

There were two main approaches we took -

1. Latent Semantic Indexing
2. Latent Dirichlet Allocation

Latent Semantic Indexing

Latent semantic analysis uses SVD to decompose the term-document matrix in the following way:

$$Term \times document = term \times topic * topic \times topic * topic \times topic \times document$$

Where term-topic matrix shows word distribution in topics, topic-topic matrix weights importance to topics, and topic-document matrix shows the topic distribution over documents.

Document similarity works better here because dimensionality reduction removes the noise in the dataset.

Latent Dirichlet Allocation

LDA functions on the idea that a document is a distribution of topics, and each topic is a distribution over words. It optimizes theta (topic distribution of a given document) and beta (word distribution over a given topic) based on given hyper-parameters of alpha and eta (which are apriori assumed probabilities of topic and word distribution respectively). We took these to be the default: i.e. starting out with uniform apriori probabilities.

This is what LDA does -

1. Assigns topics randomly to words
2. Iterates through words
3. Assume everything except current word allocation is correct. Then assigns it to a new topic based on $max(P(Z|W,D))$ where Z: topic, W: word, D: document. This value can be found using Bayes rule.
4. Repeats this until the model converges to a stable allocation. In essence, until we have found a good beta, theta.

Theta gives us topic distribution over each document, which means that given a document, we only need to look into theta to find its topic. We can choose the topic associated with maximum probability i.e. maximum contribution for the purpose of putting the emails into different clusters.

We used the coherence metric to evaluate the models that we ran.

Topic Coherence

There are multiple measures to calculate topic coherence. We decided to use the two most popular metrics -

1. c_v: sliding window to find 'similarity' between high probability words in the topic. Similarity is calculated using cosine similarity and NPMI (normalized pointwise mutual information) [0,1]
2. u_mass: document co-occurrence counts, bigram window, conditional probability [-14,+14]

Jumping ahead, the superiority of LDA over LSA was corroborated by the coherence values we found for each, for the corresponding number of clusters.

Topic Modelling Implementation

Preprocessing

The original Dataset itself was a csv with directory, email pairs. Since we did not need the directories, we discarded this information and created a dataset which contained only the Email content.

After going through a couple of Emails manually, we saw a pattern in how they were structured. Based on this, we extracted the "To", "From", and "Subject" fields of each email along with the Email body.

After this, we took this dataset and using nltk, removed stopwords from each of these emails. We also used nltk Part of Speech Tagging and WordNet Lemmatizer to tag all the words from the Email bodies and then lemmatize it accordingly. This was the preliminary dataset that we decided to work with.

Changes Made after Initial Run

With this dataset, we trained our LDA model on all the 0.5 million emails and inspected the coherence values as well as the word distributions that we got for the topics obtained. We observed that the coherence values that we were getting were in the 0.3 to 0.4 range and also that the word distributions for each topic included a lot of names and words such as "ect", "hou", "www", "http", "best", "thank" etc in high probability slots.

We manually inspected our dataset once again and realized that whenever a user forwarded an email, G-mail automatically added a couple of keywords such as "hou", "www", "http" etc. Moreover, words like "best", "thank" etc. were occupying high probability slots for all the topics, and therefore we decided that they were creating noise for the distinctiveness of the topics. We also verified that other people who had worked on Enron dataset similarly removed neutral words to increase the discriminatory power of the topics. Hence, we added these 5 words to our stop-words as well.

To counter the frequent occurrence of names in the topics, we used the “To” and “From” fields from each email to construct a dictionary of names in the dataset. We used this dictionary itself to discard names from our dataset.

The dataset we got after all these steps was used for final training.

Training

We trained the models with default parameters with the number of topics in the range 3 to 9 and every second number in the range 11 to 29, and calculated the coherence values of the trained models. Fig. 1 plots the coherence values corresponding to the number of topics the model was trained to form.

Coherence score is a measure of how often two words w_i, w_j for all $i, j \in t$, (t is the total number of words we are considering for each topic) appear together in the dataset relative to how often only one of them appears in the dataset. The scores lie between 0 and 1 and a higher score means the learned topics are good - the words in the topics tend to appear together around each other a lot. However, the score can neither be too close to 0 as that would mean that the modeled topics are not based on any words from the vocabulary nor be too close to 1 as that would imply that the dataset only has one or two words repeated throughout. A decent coherence value to aim for a learned model would be between 0.5 and 0.6.

Our third run gave coherence values between 0.496 (27 topics) and 0.562 (7 topics) and the values were in the same margin for 3, 4, 7, and 8 topics.

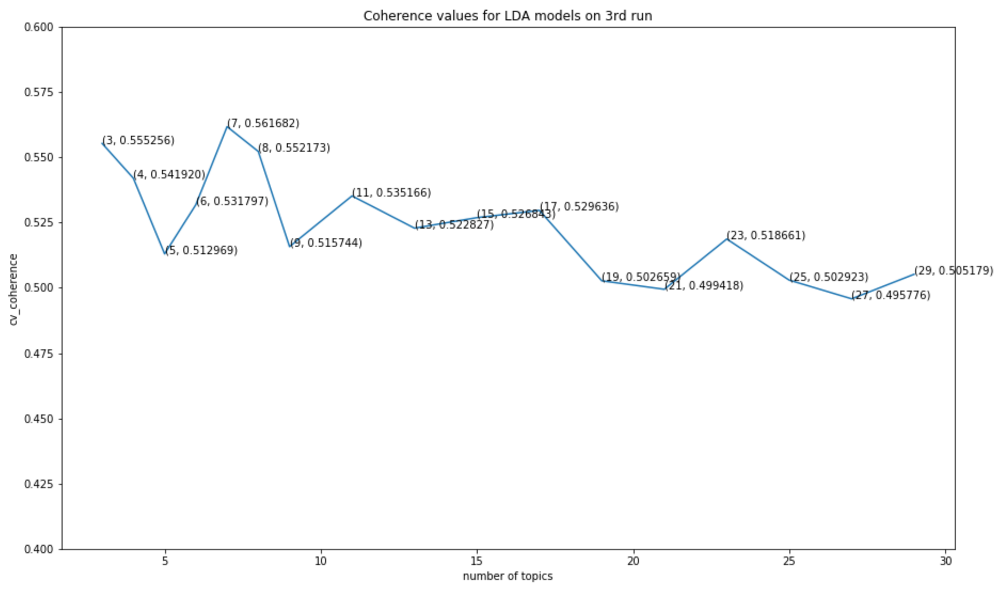


Figure 1: Coherence scores of LDA

We then trained LSA models for topic numbers 3 to 9 and the coherence values were between 0.431 (9 topics) and 0.593 (3 topics). Fig. 2 is a plot of the coherence scores of all LSA models trained in our third run.

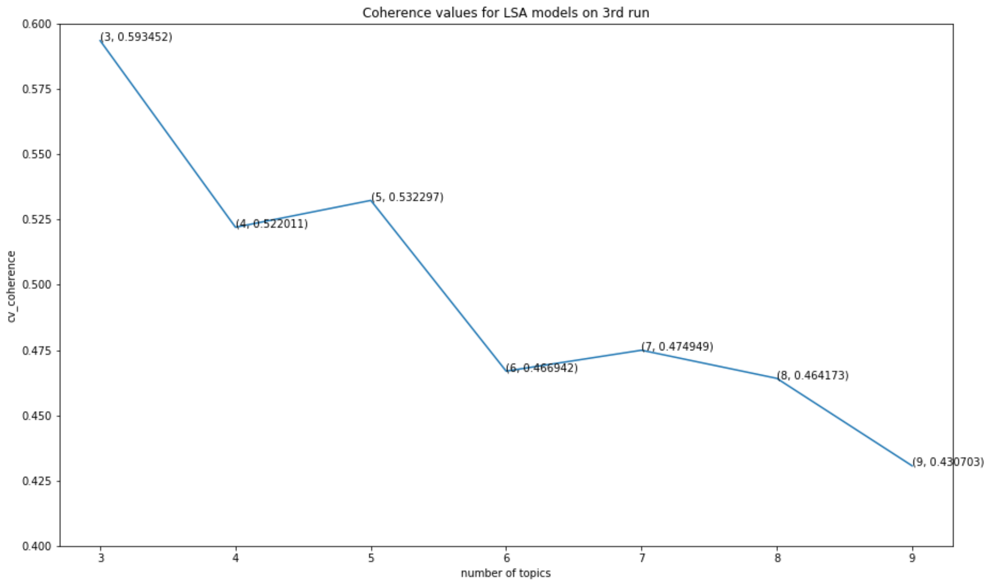
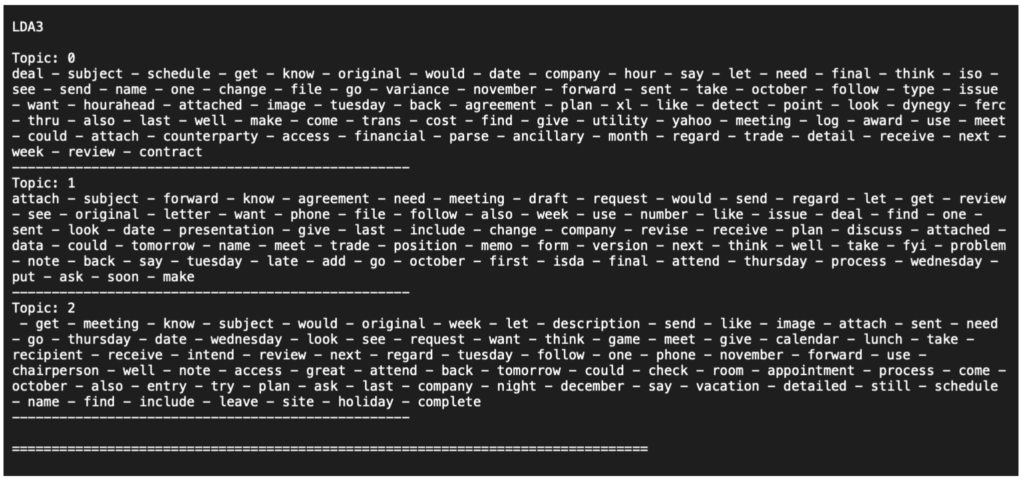


Figure 2: Coherence scores of LSA

We did some research about LDA and LSA as approaches to topic modelling and found that LDA is generally preferred. Interpret-ability is more difficult with LSA, the reduced dimensions have to be interpreted. More importantly, LSA assumes that ‘topics’ will be orthogonal i.e. it assumes that the SVD dimensions yields topics. The superiority of LDA over LSA was corroborated by the coherence values that we obtained; we found lower coherence values for LSA for almost all corresponding values of topics. We decided to continue working with the LDA model and for possible ways of improving it.

Although we got a high coherence value for 3, we can see that the topics yielded by the model are very loose or generic (this will still give high coherence values, according to way coherence is calculated). In general, we read that with LDA we are looking for a sweet-spot between topics that are too generic and too specific. Therefore we observe that 7 is the best model to choose.

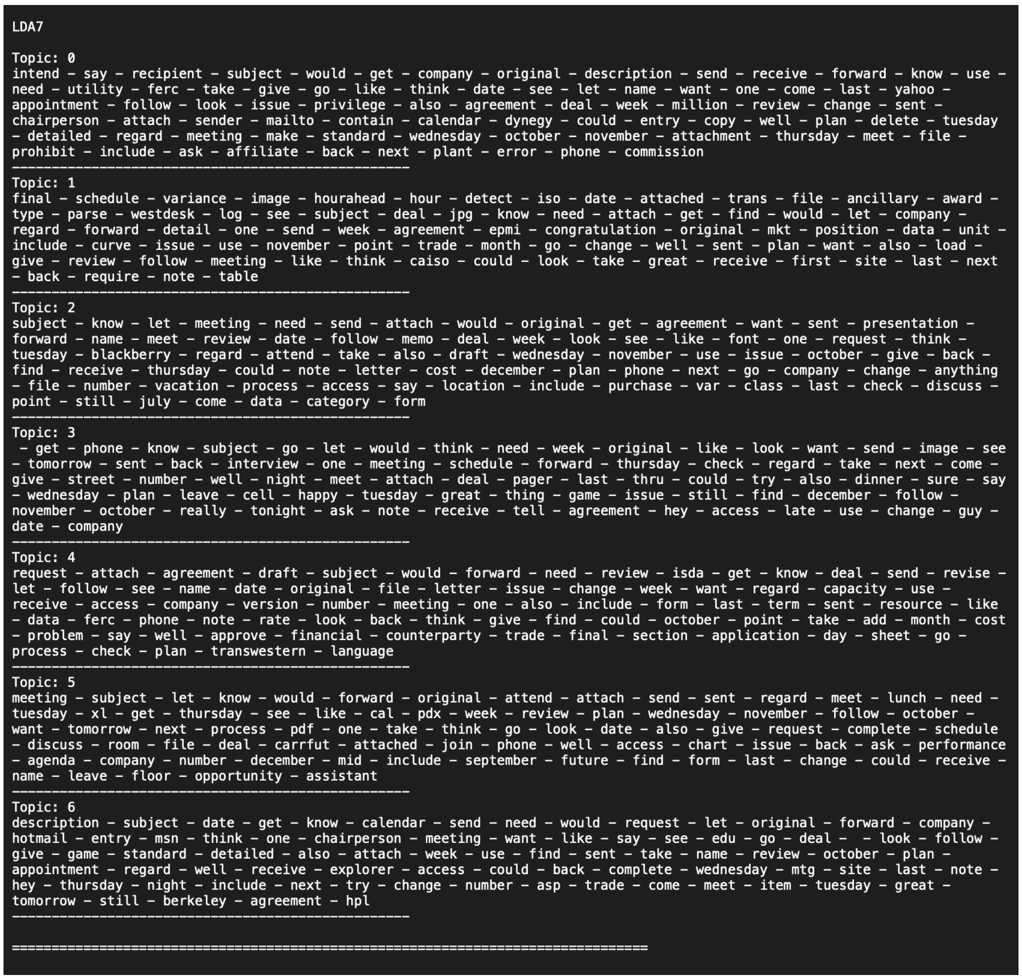
Fig. 3 shows the top 80 words in decreasing order of their probability of being in that topic for the three topics that the model came up with when the number of topics was set to 3. We can see that even though the coherence value is high for this, there is a lot of overlap between the topics (similar words appearing with high probability for different topics).



```
LDA3

Topic: 0
deal – subject – schedule – get – know – original – would – date – company – hour – say – let – need – final – think – iso –
see – send – name – one – change – file – go – variance – november – forward – sent – take – october – follow – type – issue
– want – hourahead – attached – image – tuesday – back – agreement – plan – xl – like – detect – point – look – dynegy – ferc
– thru – also – last – well – make – come – trans – cost – find – give – utility – yahoo – meeting – log – award – use – meet
– could – attach – counterparty – access – financial – parse – ancillary – month – regard – trade – detail – receive – next –
week – review – contract
--------------------------------------------------
Topic: 1
attach – subject – forward – know – agreement – need – meeting – draft – request – would – send – regard – let – get – review
– see – original – letter – want – phone – file – follow – also – week – use – number – like – issue – deal – find – one –
sent – look – date – presentation – give – last – include – change – company – revise – receive – plan – discuss – attached –
data – could – tomorrow – name – meet – trade – position – memo – form – version – next – think – well – take – fyi – problem
– note – back – say – tuesday – late – add – go – october – first – isda – final – attend – thursday – process – wednesday –
put – ask – soon – make
--------------------------------------------------
Topic: 2
 – get – meeting – know – subject – would – original – week – let – description – send – like – image – attach – sent – need
– go – thursday – date – wednesday – look – see – request – want – think – game – meet – give – calendar – lunch – take –
recipient – receive – intend – review – next – regard – tuesday – follow – one – phone – november – forward – use –
chairperson – well – note – access – great – attend – back – tomorrow – could – check – room – appointment – process – come –
october – also – entry – try – plan – ask – last – company – night – december – say – vacation – detailed – still – schedule
– name – find – include – leave – site – holiday – complete
--------------------------------------------------


================================================================================
```

Figure 3: Topics and their word distributions for LDA3

Fig. 4 shows the top 80 words with decreasing probabilities of appearing in the topics learned by an LDA for 7 topics. We can notice that the difference between topics is better encapsulated in this model and the overlap between topics is much lower than what we saw in LDA for 3 topics.



```
LDA7

Topic: 0
intend - say - recipient - subject - would - get - company - original - description - send - receive - forward - know - use -
need - utility - ferc - take - give - go - like - think - date - see - let - name - want - one - come - last - yahoo -
appointment - follow - look - issue - privilege - also - agreement - deal - week - million - review - change - sent -
chairperson - attach - sender - mailto - contain - calendar - dynegy - could - entry - copy - well - plan - delete - tuesday
- detailed - regard - meeting - make - standard - wednesday - october - november - attachment - thursday - meet - file -
prohibit - include - ask - affiliate - back - next - plant - error - phone - commission
--------------------------------------------------
Topic: 1
final - schedule - variance - image - hourahead - hour - detect - iso - date - attached - trans - file - ancillary - award -
type - parse - westdesk - log - see - subject - deal - jpg - know - need - attach - get - find - would - let - company -
regard - forward - detail - one - send - week - agreement - epmi - congratulation - original - mkt - position - data - unit -
include - curve - issue - use - november - point - trade - month - go - change - well - sent - plan - want - also - load -
give - review - follow - meeting - like - think - caiso - could - look - take - great - receive - first - site - last - next
- back - require - note - table
--------------------------------------------------
Topic: 2
subject - know - let - meeting - need - send - attach - would - original - get - agreement - want - sent - presentation -
forward - name - meet - review - date - follow - memo - deal - week - look - see - like - font - one - request - think -
tuesday - blackberry - regard - attend - take - also - draft - wednesday - november - use - issue - october - give - back -
find - receive - thursday - could - note - letter - cost - december - plan - phone - next - go - company - change - anything
- file - number - vacation - process - access - say - location - include - purchase - var - class - last - check - discuss -
point - still - july - come - data - category - form
--------------------------------------------------
Topic: 3
 - get - phone - know - subject - go - let - would - think - need - week - original - like - look - want - send - image - see
- tomorrow - sent - back - interview - one - meeting - schedule - forward - thursday - check - regard - take - next - come -
give - street - number - well - night - meet - attach - deal - pager - last - thru - could - try - also - dinner - sure - say
- wednesday - plan - leave - cell - happy - tuesday - great - thing - game - issue - still - find - december - follow -
november - october - really - tonight - ask - note - receive - tell - agreement - hey - access - late - use - change - guy -
date - company
--------------------------------------------------
Topic: 4
request - attach - agreement - draft - subject - would - forward - need - review - isda - get - know - deal - send - revise -
let - follow - see - name - date - original - file - letter - issue - change - week - want - regard - capacity - use -
receive - access - company - version - number - meeting - one - also - include - form - last - term - sent - resource - like
- data - ferc - phone - note - rate - look - back - think - give - find - could - october - point - take - add - month - cost
- problem - say - well - approve - financial - counterparty - trade - final - section - application - day - sheet - go -
process - check - plan - transwestern - language
--------------------------------------------------
Topic: 5
meeting - subject - let - know - would - forward - original - attend - attach - send - sent - regard - meet - lunch - need -
tuesday - xl - get - thursday - see - like - cal - pdx - week - review - plan - wednesday - november - follow - october -
want - tomorrow - next - process - pdf - one - take - think - go - look - date - also - give - request - complete - schedule
- discuss - room - file - deal - carrfut - attached - join - phone - well - access - chart - issue - back - ask - performance
- agenda - company - number - december - mid - include - september - future - find - form - last - change - could - receive -
name - leave - floor - opportunity - assistant
--------------------------------------------------
Topic: 6
description - subject - date - get - know - calendar - send - need - would - request - let - original - forward - company -
hotmail - entry - msn - think - one - chairperson - meeting - want - like - say - see - edu - go - deal -  - look - follow -
give - game - standard - detailed - also - attach - week - use - find - sent - take - name - review - october - plan -
appointment - regard - well - receive - explorer - access - could - back - complete - wednesday - mtg - site - last - note -
hey - thursday - night - include - next - try - change - number - asp - trade - come - meet - item - tuesday - great -
tomorrow - still - berkeley - agreement - hpl
--------------------------------------------------


================================================================================
```

Figure 4: Topics and their word distributions for LDA7

Model Improvement

In our third run, we ran everything with default values to look for differences between LDA and LSA. Once we decided to continue with LDA, there was more room for coherence score improvement. Initially, the chunk size (γ) was set to 2000 - this meant that before every batch update, the algorithm looked at a sample of 2000 data points and assumed the topic allocation for the who sample of 2000 data points and assumed the topic allocation for the whole dataset (λ) to be correct. It then updated the topics (and word allocations) for the 2000 emails (γ) - what has been referred to as the E-step (Hoffman et al. (2010)) if we draw an analogy between batch learning LDA to an Expectation-Maximisation Algorithm (Dempster et al. (1977)) - and then updated the topics (λ) themselves, given word allocations (ϕ) - the M-step.

Lower lambda means less data points are loaded to memory before every update but the context (in strict terms of the number of data points) of the sample in which the updates are made are low too. We decided to increase λ to approximately 10% of the size of the dataset. It would look at larger batches of documents when updating the topics of those documents before updating the to sample in which the updates are made are low too. With this increased value, the training would look at larger batches of documents when updating the topics of those documents before updating the topics. However, a larger chunk size means less batch updates which is why we increased the number of passes - the number of times the algorithm will pass over the entire dataset during training. We set number of passes to 2 and chunk size to 51740 before our last run.

Fig. 5 compares the coherence values we got in the third run with the values after changing the hyper-parameters (fourth run).

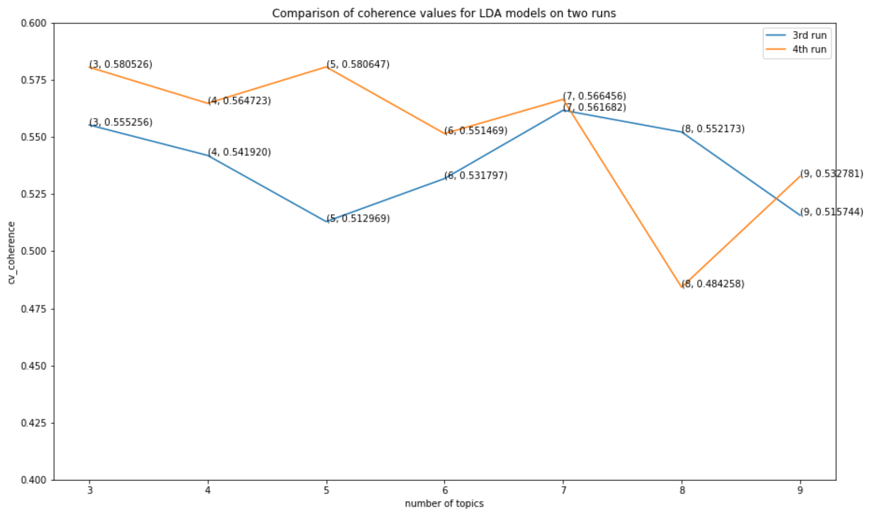


Figure 5: Comparison of coherence values before and after changing chunk_size

We retained the models from both the runs and looked at other metrics for comparison. One of them was perplexity values for each of those models. Perplexity is a measure of how well a probabilistic model predicts a sample and is calculated as the normalised log-likelihood of a held out test set. Lower perplexity is better. Fig. 6 shows a comparison of perplexity values for LDA before and after changing the hyper-parameters when topic numbers were set from 3 to 9.

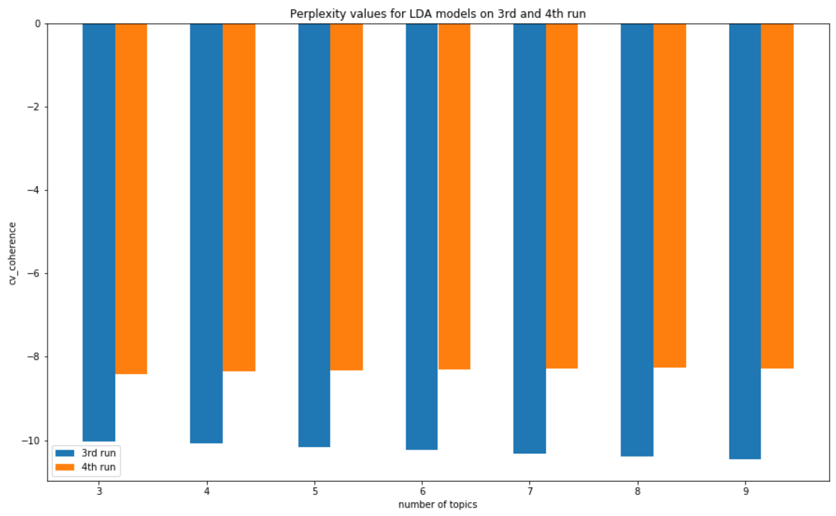


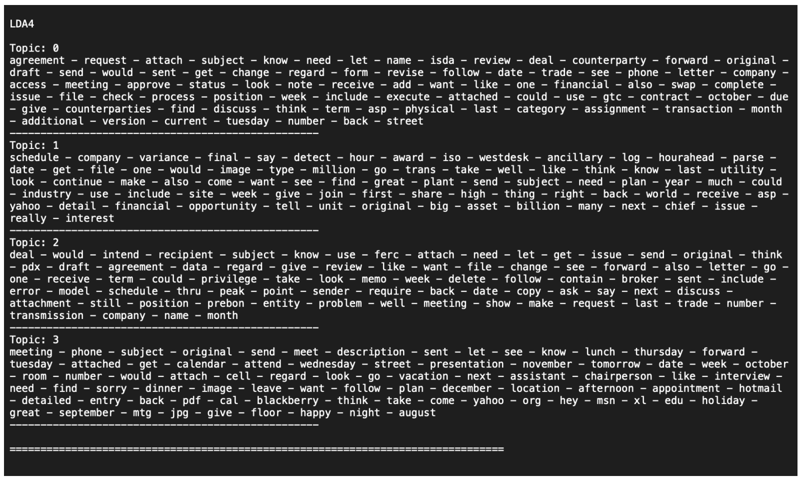
Figure 6: Comparison of perplexity scores before and after changing chunk_size

The old perplexity values range from $(-10.450, -10.038)$ and after changing the hyper-parameters, the perplexity values are in the range $(-8.408, -8.251)$. The lowest perplexity score after the last run was -8.408 (for 3 topics) but again, three is a simplistic clustering and the topics did not make sense. Perplexity values when number of topics are 4 and 5 are -8.34 and -8.33 respectively.

Number of Topics	Run3 Coherence	Run4 Coherence	Run3 Perplexity	Run4 Perplexity
3	0.555256	0.580526	-10.038018	-8.408472
4	0.541920	0.564732	-10.074575	-8.348842
5	0.512969	0.580647	-10.160900	-8.333169
7	0.561682	0.566456	-10.312154	-8.283350

However, it is important to note that these are just predictive likelihood measures which studies like Chang et al. (2009) have shown are not always correlated to human judgement. So even though we can look at the perplexity metric as a sign of improvement, we should pay more attention to coherence score and finally, plain old human judgement to decide on the interpret-ability of topics given by a predictive model.

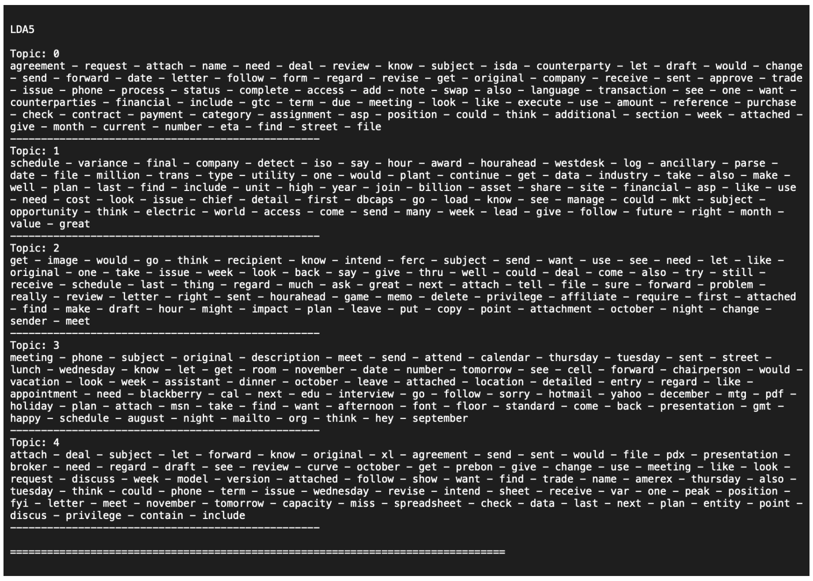
Fig. 7 lists the topic words for the model with 4 topics. The topics seem to define different things and are fairly exclusive given the dataset.



```
LDA4

Topic: 0
agreement – request – attach – subject – know – need – let – name – isda – review – deal – counterparty – forward – original –
draft – send – would – sent – get – change – regard – form – revise – follow – date – trade – see – phone – letter – company –
access – meeting – approve – status – look – note – receive – add – want – like – one – financial – also – swap – complete –
issue – file – check – process – position – week – include – execute – attached – could – use – gtc – contract – october – due
– give – counterparties – find – discuss – think – term – asp – physical – last – category – assignment – transaction – month
– additional – version – current – tuesday – number – back – street
-------------------------------------------------
Topic: 1
schedule – company – variance – final – say – detect – hour – award – iso – westdesk – ancillary – log – hourahead – parse –
date – get – file – one – would – image – type – million – go – trans – take – well – like – think – know – last – utility –
look – continue – make – also – come – want – see – find – great – plant – send – subject – need – plan – year – much – could
– industry – use – include – site – week – give – join – first – share – high – thing – right – back – world – receive – asp –
yahoo – detail – financial – opportunity – tell – unit – original – big – asset – billion – many – next – chief – issue –
really – interest
-------------------------------------------------
Topic: 2
deal – would – intend – recipient – subject – know – use – ferc – attach – need – let – get – issue – send – original – think
– pdx – draft – agreement – data – regard – give – review – like – want – file – change – see – forward – also – letter – go –
one – receive – term – could – privilege – take – look – memo – week – delete – follow – contain – broker – sent – include –
error – model – schedule – thru – peak – point – sender – require – back – date – copy – ask – say – next – discuss –
attachment – still – position – prebon – entity – problem – well – meeting – show – make – request – last – trade – number –
transmission – company – name – month
-------------------------------------------------
Topic: 3
meeting – phone – subject – original – send – meet – description – sent – let – see – know – lunch – thursday – forward –
tuesday – attached – get – calendar – attend – wednesday – street – presentation – november – tomorrow – date – week – october
– room – number – would – attach – cell – regard – look – go – vacation – next – assistant – chairperson – like – interview –
need – find – sorry – dinner – image – leave – want – follow – plan – december – location – afternoon – appointment – hotmail
– detailed – entry – back – pdf – cal – blackberry – think – take – come – yahoo – org – hey – msn – xl – edu – holiday –
great – september – mtg – jpg – give – floor – happy – night – august
-------------------------------------------------


==============================================================================
```

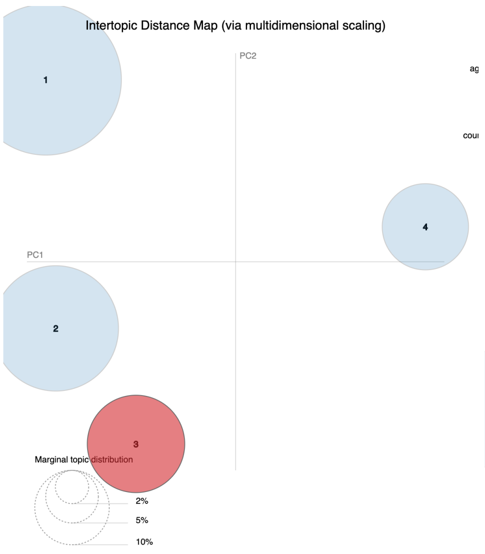
Figure 7: Topics and their word distributions for LDA4 (after improvement)

Fig. 8 is a list of top 60 topic words that the model with 5 topics produces. Although these topics do define distinct things just like the previous one, there is some overlap between topic0 and topic4. From the looks of it, it decomposes one of the topics into two based on more subtle differences.

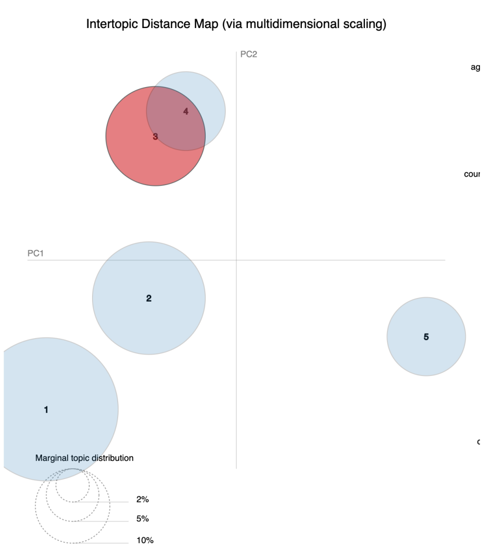


```
LDA5

Topic: 0
agreement – request – attach – name – need – deal – review – know – subject – isda – counterparty – let – draft – would – change
– send – forward – date – letter – follow – form – regard – revise – get – original – company – receive – sent – approve – trade
– issue – phone – process – status – complete – access – add – note – swap – also – language – transaction – see – one – want –
counterparties – financial – include – gtc – term – due – meeting – look – like – execute – use – amount – reference – purchase
– check – contract – payment – category – assignment – asp – position – could – think – additional – section – week – attached –
give – month – current – number – eta – find – street – file
-------------------------------------------------
Topic: 1
schedule – variance – final – company – detect – iso – say – hour – award – hourahead – westdesk – log – ancillary – parse –
date – file – million – trans – type – utility – one – would – plant – continue – get – data – industry – take – also – make –
well – plan – last – find – include – unit – high – year – join – billion – asset – share – site – financial – asp – like – use
– need – cost – look – issue – chief – detail – first – dbcaps – go – load – know – see – manage – could – mkt – subject –
opportunity – think – electric – world – access – come – send – many – week – lead – give – follow – future – right – month –
value – great
-------------------------------------------------
Topic: 2
get – image – would – go – think – recipient – know – intend – ferc – subject – send – want – use – see – need – let – like –
original – one – take – issue – week – look – back – say – give – thru – well – could – deal – come – also – try – still –
receive – schedule – last – thing – regard – much – ask – great – next – attach – tell – file – sure – forward – problem –
really – review – letter – right – sent – hourahead – game – memo – delete – privilege – affiliate – require – first – attached
– find – make – draft – hour – might – impact – plan – leave – put – copy – point – attachment – october – night – change –
sender – meet
-------------------------------------------------
Topic: 3
meeting – phone – subject – original – description – meet – send – attend – calendar – thursday – tuesday – sent – street –
lunch – wednesday – know – let – get – room – november – date – number – tomorrow – see – cell – forward – chairperson – would –
vacation – look – week – assistant – dinner – october – leave – attached – location – detailed – entry – regard – like –
appointment – need – blackberry – cal – next – edu – interview – go – follow – sorry – hotmail – yahoo – december – mtg – pdf –
holiday – plan – attach – msn – take – find – want – afternoon – font – floor – standard – come – back – presentation – gmt –
happy – schedule – august – night – mailto – org – think – hey – september
-------------------------------------------------
Topic: 4
attach – deal – subject – let – forward – know – original – xl – agreement – send – sent – would – file – pdx – presentation –
broker – need – regard – draft – see – review – curve – october – get – prebon – give – change – use – meeting – like – look –
request – discuss – week – model – version – attached – follow – show – want – find – trade – name – amerex – thursday – also –
tuesday – think – could – phone – term – issue – wednesday – revise – intend – sheet – receive – var – one – peak – position –
fyi – letter – meet – november – tomorrow – capacity – miss – spreadsheet – check – data – last – next – plan – entity – point –
discus – privilege – contain – include
-------------------------------------------------


==============================================================================
```

Figure 8: Topics and their word distributions for LDA5 (after improvement)



(a) LDA4



(b) LDA5

Figure 9: Visualisation of final models

The interactive visualizations can be found here - https://checker5965.github.io/vizlda4 and https://checker5965.github.io/vizlda5. If we look at the visualisation of a multidimensional scaling of both the models in fig. 9a and fig. 9b, we get a better sense of how these topics differ. Something that we observed over many trials (over randomly picked emails) was that lda5 predicted the highest topic distribution with more certainty: i.e. the difference in its probability for the highest topic and the second highest topic is much larger as compared to that of lda4. We interpreted this to mean that lda5 has more discriminatory power than lda4.

As a final use of the model, we pick some sample email and assign a topic to it. In table 2, we have added the links to the output files of a few emails which are hosted on an independent server. In the results file for each email, we have included (in this order) the original email, the output from the pre-processor (this serves as an input for the training algorithm), the topic probability distribution for that email, the assigned topic (topic with highest probability in the distribution we receive from the model) and the remaining unassigned topics with decreasing probability along with the word distributions of the respective topics in the results file.

Email ID	Topic Assigned (LDA 5)
68	Email 68 topics
23589	Email 23589 topics
129302	Email 129302 topics
37648	Email 37648 topics

Conclusion and Future Work

There are certain things that we can look at in the future to improve our approach. At the pre-processing stage, we could remove the privacy disclaimers at the bottom of certain random emails (this is something that other people who have done LDA have done, arguing that the presence of the disclaimer has nothing to do with whether the email has confidential content, is quite arbitrary, and all it does is adds more ground for unrelated emails to be linked by that text.) We discussed that representing documents using TF-IDF values for K-Means clustering is not very good: we can look into doc2vec representations for the same. In conclusion, we obtained an optimal LDA model with 5 topics to assign topic distributions to each document in the dataset, which helps us structure the content of these emails.

References

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Chang, J., Boyd-Graber, J., Gerrish, S., Wang, C., and Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Proceedings of the 22Nd International Conference on Neural Information Processing Systems*, NIPS'09, pages 288–296, USA. Curran Associates Inc.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Hermans, F. and Murphy-Hill, E. (2015). Enron's spreadsheets and related emails: A dataset and analysis. In *Proceedings of the 37th International Conference on Software Engineering - Volume 2*, ICSE '15, pages 7–16, Piscataway, NJ, USA. IEEE Press.

Hoffman, M. D., Blei, D. M., and Bach, F. (2010). Online learning for latent dirichlet allocation. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, pages 856–864, USA. Curran Associates Inc.

Kaelbling, L. and Gervasio, M. (2015). The enron corpus. `https://www.cs.cmu.edu/~enron/`.

Kumar, H. and Aldous, D. (2015). Exploratory data analysis of enron emails. *University of California-Berkeley.*

Repke, T., Krestel, R., Edding, J., Hartmann, M., Hering, J., Kipping, D., Schmidt, H., Scordialo, N., and Zenner, A. (2018). Beacon in the dark: A system for interactive exploration of large email corpora. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pages 1871–1874, New York, NY, USA. ACM.

Sappelli, M., Pasi, G., Verberne, S., de Boer, M., and Kraaij, W. (2016). Assessing e-mail intent and tasks in e-mail messages. *Information Sciences*, 358-359:1 – 17.