LDA

What we did

Why we chose it

What we expected

What we actually got out

# LDA

## What we did:

One method that is common for text analysis is Latent Dirichlet Allocation (LDA). This method groups documents into topics. The results of an LDA is a probabilistic grouping of documents based the terms in the entire set of documents.

The jobs data was cleaned to remove extraneous characters and to remove terms which we did not want to be used in the category. Specifically, our business purpose is to rank jobs based on job similarity, so the name of the company was removed from the text of the document. Otherwise it was likely that the name of the company repeated multiple times in a document would become a strong influence in the document breakdown in this analysis. Additionally, the standard text cleaning methods of converting to lowercase, removing punctuation, and cleaning whitespace were performed. The corpus was then converted to a document term matrix (DTM).

Next, an LDA was run on the DTM. The first model run was with the number of topics set at k=5. The next model was at k=50. Due to the time consuming nature of running the LDA, these were the only allocations that were run. The top 10 terms for each lda was produced to get a representation of the major topics in each allocation. Next the topic probabilities were calculated to identify the increased likelihood of a topic based on each term. Additionally, term probabilities were calculated to identify the probability of each term occurring in each of the topics.

The resumes were loaded into a document term process using the same clean-up and process that was used to prepare the jobs data for the LDA. Next, the topic likelihoods and term likelihoods were calculated for the resumes using the posterior() function to calculate based on the previous LDA. The most likely topic for each resume was also calculated.

The next analysis that was planned was to use the LDA to identify the most relevant words (based on a not-yet identified cut-off) for that document. Then we anticipated that we would be able to search resumes for those particular terms only and perform a distance measure from each topic. Additionally, it was anticipated that an analysis would be performed to identify how much of the data was covered by the LDA. Currently it is unknown whether the model is underfit or overfit. Further, more analysis can be performed to identify the characteristics of each topic more than just getting the top 10 characteristics.

## Why we chose it

One of the goals of this analysis is to identify jobs that a candidate is a good fit for. LDA allows for the reduction of dimensionality in the final analysis. It also produces the probabilities that words terms will exists in documents which is useful in calculating whether another document (the resume) has high likelihood of matching the job posting based on comparison of terms. One of the concerns when selecting this method was that LDA is best at categorizing documents. It may have problems with identifying when a document doesn’t fit into any of the topics and will simply select the best topic (by default). This requires additional work beyond LDA to identify how well the document/resume matches the topic. In other words, it identifies the closest topic easily, but doesn’t indicate when the document doesn’t belong in any topic.

## What we expected

The LDA was expected to help in the following ways:

1. Categorize the jobs data into topics
2. Identify important terms from each topic

## What we got

Running the LDA with k=50 resulted in a very wide topic model. A visual review (below) shows that many of the terms that are identified are not exceptionally representative of specific requirements. It is not expected that employers will look for “will” or “business” or “time” in a resume. Additionally, this model does not clearly represent the most important terms to employers. One of the interesting trends in the data is that Siri is a top term in more than one of the topics models identified. Also, machine learning makes a good showing in Topic 5.



In addition to the top terms in the document, the LDA produced the probabilities that a term would occur in a document within each topic, and the probability that document would match a topic. The output for this information is not clear enough to post here but is available in the code on GitHub.

Running the LDA with k=5 resulted in a much simpler topic model (top 10 terms below). Unfortunately, this also resulted in even less critical terms being in the top of each document. This is likely because the jobs data is full of narratives and many of the words are used multiple times. The more critical tools and processes are only listed few (or one) times and therefore are not strongly represented here.

