STAKEHOLDER REPORT

Washington Mutual: The Lost Bank.

Summary

What patterns can be found in literature about the crash of Washing Mutual. Who is connected and what predictions can be made on literature reviews?

Frederick Kjærgaard , Claus Mørkbak Højrup, Mark Daniel von Kelaita and Martin Jæger Nielsen

Introduction:

In the financial crisis of 2008, Washington Mutual Bank (also known as WaMu) failed. The bank was not able to absorb losses and meet demands when customers stopped paying back loans. In addition, word got out that Washington Mutual was on shaky financial ground, so customers with money at WaMu made a run on the bank. They pulled out as much money, as they could, which made it even harder for Washington Mutual to survive. As a result of the failure, Washington Mutual's assets were sold to Chase Bank.

The collapse of Washington Mutual in September 2008 was the largest bank failure in U.S. history and a symbolic casualty of America's unfolding financial crisis.

Description of the data:

This report is based on data extracted from the book "The Lost Bank". The book can be purchased here: https://www.amazon.com/Lost-Bank-Washington-Mutual-American/dp/B00CC6EHRC

The book is written by Wall Street Journal reporter Kristen Grind, who provides an account of the bank's final hours. Grind has already received numerous awards for her coverage of WaMu's fall and is considered one of the best sources of this piece of history.

The data is extracted from the book and loaded into a dataframe with each sentence in its own row. Initital cleaning includes stopwords, weird characters and lemmatizing.

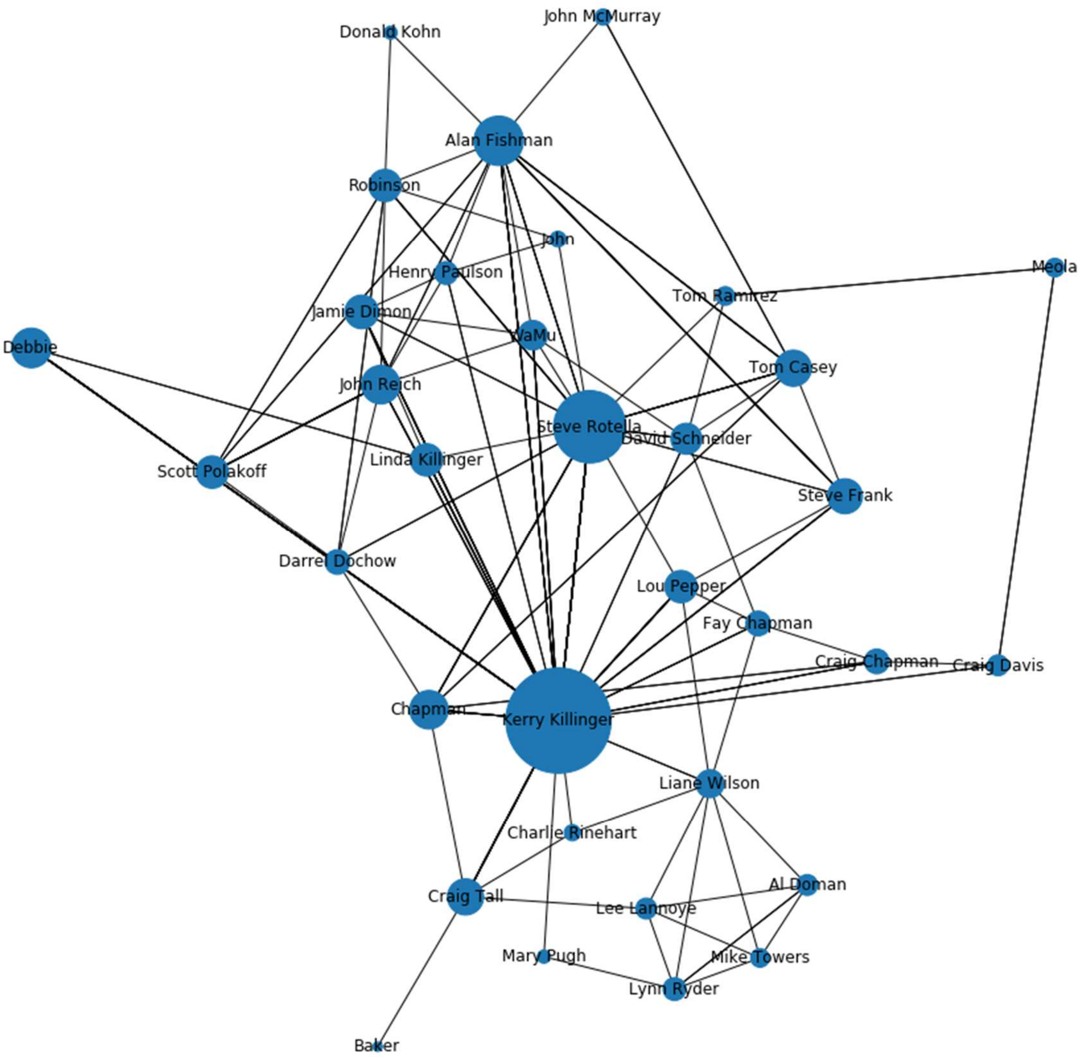
Workflow structure:

After cleaning of the data, we structure a workflow as follows:

1) Edge and nodelists with the purpose of visualizing a network and extracting information about the major players around the crash of WaMu.
2) NLP and clustering with the purpose of making topics and patterns visible.
3) Webscraping of Amazons reviews of the sourcematerial. What do the customer say about the book? This part is not based on material from the base source (the book) but on material about the book and is directed toward making classification of reviews about the book.

Uncovering network based on the data:

To get an understanding of the book and how characters relate to each other, we create a network based on when the characters appear in the same sentence. We do this using space identify to persons and organizations in each sentence. We sort out the sentences which only contain one element, since they aren't usable for creating a network



The size of the nodes represents the degree centrality of each node, which is the amount of connections relative to the maximum amount of connection observed. The thickness of each line represents the amount of edges between each node. This is the amount of times two persons are mentioned in the same sentence. Our network has 34 nodes are 238 edges/connections. These nodes are have a density of 0.1639 and a Transitivity of 0.3493. The Transitivity is the mount of possible nodes of triangles exist in the network. A third of the people who are connected to one person are also connected with each other.

```
[ ]  #Look at the top 5 nodes by the amount of unique edges
     N_unique=sorted(H.degree, key=lambda x: x[1], reverse=True)
     print(N_unique[0:5])

     [('Kerry Killinger', 21), ('Steve Rotella', 16), ('Alan Fishman', 11), ('Liane Wilson', 8), ('David Schneider', 8)]

[ ]  #Look at the top 5 nodes by the amount of total edges
     N_common=sorted(G.degree, key=lambda x: x[1], reverse=True)
     print(N_common[0:5])

     [('Kerry Killinger', 127), ('Steve Rotella', 60), ('Alan Fishman', 28), ('Debbie', 18), ('John Reich', 17)]

[ ]  #Look at the top 9 most common edges.
     Counter(G.edges()).most_common(9)

     [(('Steve Rotella', 'Kerry Killinger'), 25),
      (('Debbie', 'Kerry Killinger'), 16),
      (('Kerry Killinger', 'Craig Tall'), 11),
      (('Jamie Dimon', 'Kerry Killinger'), 8),
      (('Lou Pepper', 'Kerry Killinger'), 8),
      (('Kerry Killinger', 'Linda Killinger'), 8),
      (('John Reich', 'Scott Polakoff'), 7),
      (('Alan Fishman', 'Steve Frank'), 6),
      (('Alan Fishman', 'Kerry Killinger'), 6),
      (('Chapman', 'Kerry Killinger'), 6)]
```

From this network it is clear that Kerry Killinger is the most central figure in the book, followed by Steve Rotella and Alan Fishman.

- Kerry Killinger previously served as chairman and chief executive officer of Washington-Mutual until September 8, 2008 where he was let go.
- Steve Rotella was the president and chief operating officer of Washington-Mutual during the events of the book.
- Alan Fishman was the chief executive officer of Washington-Mutual during during the events of the book.

Uncovering patterns and topics in the data:

Pre-process – Cleaning an optimizing data

In order to get a further understanding of whether there are any patterns and topics in the text, we applied Unsupervised Machine Learning with Natural Language Processing through LSI & LDA models. Before these steps could be reached, we needed to clean our data through multiple pre-processing procedures. The purpose of the pre-process is to enhance the data before it is used to perform the ML part and hereby achieving better results. The cleaning process removes unwanted words that only is seen as applying noise to the analysis. Furthermore, the unwanted words also don’t apply any meaning structure to the context of which the text comes from. In general, the pre-process included putting all text into lower case and removing all kinds of punctuations, stop words, symbols, digits and verbs. Furthermore, the text was also lemmatized and tagged for specified Part-of-speech to remove unwanted words. The purpose of lemmatizing is to take the words to its root form but keeps the context. This is the reason why lemmatizing is seen as far superior compared to stemming. Finally, the data was word tokenized. The tokenizing is mandatory for the further processing as it makes it possible to create a matrix in the form of Bag-of-Words (BOW) and TF-IDF. The TF-IDF will be used to identify topics later on.

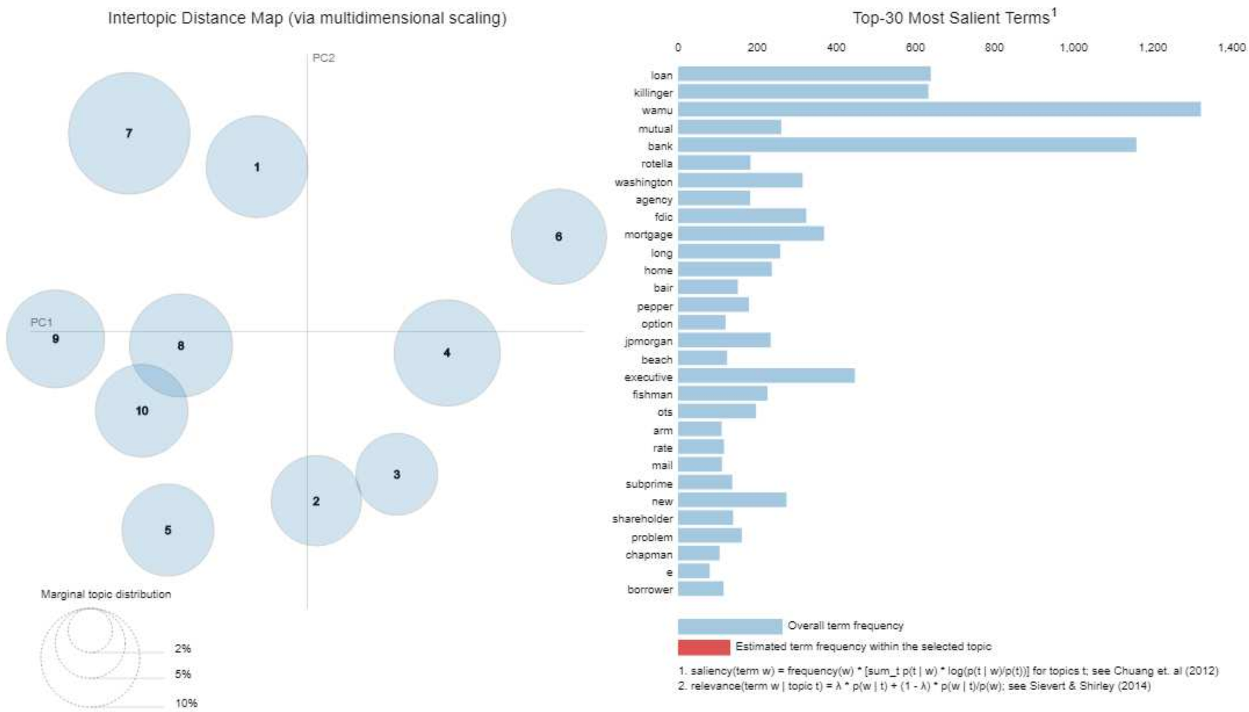
LDA (Latent Dirichlet Allocation) - Model

The LDA-Model is an approach that creates different topics based on the word frequency from a set of text. LDA are therefore good to find mixtures of topics within text/documents. Regarding the underlying techniques/coding in this approach will not be considered in this report. We chose to let the machine find 10 different topics clusters. The output of each of these individual topics is made of the frequency of different word with different influence. In our case regarding topic, we could see some patterns appearing from the different clusters. For example, that Killinger and Rotella, the two top leaders in bank, are appearing in many of the clusters, which lead us to that they play a big part of the text. The same goes with WaMu as mention before is the name of the bank. We can also identify a pattern in for instance, topic 8 for example that about the bad financial state of the bank. These words circle around different kinds of loans related to size. In comparison topic 9 contains words that could be related to the people and/or organisations involved of the crisis.

```
# Check out topics looks a bit like clusters!
lda_model.print_topics(-1)
```

```
[(0,
  '0.039*"washington" + 0.037*"mutual" + 0.022*"option" + 0.021*"wamu" + 0.019*"arm" + 0.013*"change" + 0.013*"bank" + 0.012*"news" + 0.011*"branch" + 0.010*"loan"'),
 (1,
  '0.034*"killinger" + 0.031*"bair" + 0.021*"fishman" + 0.020*"problem" + 0.020*"mail" + 0.017*"big" + 0.016*"e" + 0.016*"bank" + 0.015*"wamu" + 0.011*"chief"'),
 (2,
  '0.016*"call" + 0.016*"wall" + 0.015*"street" + 0.014*"well" + 0.014*"also" + 0.013*"fishman" + 0.011*"customer" + 0.010*"reich" + 0.010*"john" + 0.009*"paulson"'),
 (3,
  '0.055*"killinger" + 0.042*"executive" + 0.026*"new" + 0.026*"pepper" + 0.018*"wamu" + 0.017*"back" + 0.015*"york" + 0.014*"people" + 0.013*"team" + 0.013*"office"'),
 (4,
  '0.049*"wamu" + 0.046*"bank" + 0.016*"regulator" + 0.014*"ots" + 0.014*"fishman" + 0.013*"team" + 0.013*"fdic" + 0.012*"executive" + 0.012*"capital" + 0.010*"time"'),
 (5,
  '0.036*"rotella" + 0.028*"killinger" + 0.018*"job" + 0.018*"chapman" + 0.017*"long" + 0.013*"great" + 0.013*"mutual" + 0.012*"washington" + 0.012*"house" + 0.011*"tall"'),
 (6,
  '0.078*"loan" + 0.032*"mortgage" + 0.030*"bank" + 0.030*"home" + 0.020*"high" + 0.018*"subprime" + 0.017*"rate" + 0.014*"borrower" + 0.014*"wamu" + 0.013*"percent"'),
 (7,
  '0.074*"wamu" + 0.047*"bank" + 0.023*"jpmorgan" + 0.019*"shareholder" + 0.019*"day" + 0.015*"employee" + 0.012*"year" + 0.012*"company" + 0.010*"executive" + 0.009*"team"'),
 (8,
  '0.063*"wamu" + 0.055*"bank" + 0.031*"fdic" + 0.016*"financial" + 0.013*"large" + 0.012*"ots" + 0.011*"jpmorgan" + 0.011*"deposit" + 0.011*"bad" + 0.010*"company"'),
 (9,
  '0.035*"agency" + 0.033*"wamu" + 0.033*"long" + 0.027*"bank" + 0.024*"beach" + 0.021*"fdic" + 0.019*"mortgage" + 0.018*"ots" + 0.014*"time" + 0.014*"director"')]
```

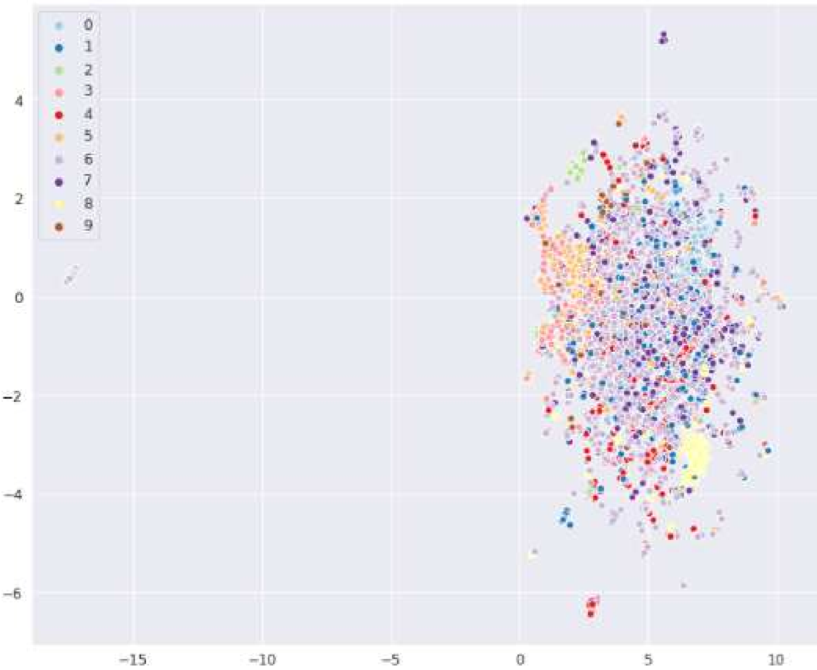
In endeavouring to get a better understanding we visualised the topics in an intertopic distance map which tries to cluster them. The visualisation shows that topic 6 is positioned near the top right corner unlike topic 5 which is placed in the bottom left. The distance represents how similar the different topic clusters are to each other. In addition, we can see that topic 8 and 10 are the topics closest to topic 9 and therefore should be more like this, which more or less can be verified by looking at their individual words.



By looking at the overall result from the intertopic distance map we can see some of the most frequent words are the following "Loan, Killinger, WaMu, subprime, Bank, executives, Rotella". This concludes that from LDA we can identify some of the main reason for bad performance in the bank, and who is involved.

LSI – Model

The LSI-Model model takes all the text in the matrix and performs a decomposition (SVD) on the term-document matrix, this way the model tries to identify patterns and thereby topics. Training the LSI is faster than the LDA, all though it is less accurate and harder to interpret. The LSI allowed us to make some similarity-queries that tells us the which of the documents are most similar to each other regarding topic identification. In the LSI we see similar result regarding topics. Here we also see that "Bank, WaMu, Killinger, Loan" is appearing in many of the cluster



topics. But we also see some new like how customers also are involved. This can be deduced from the result below.

Though we also applied K-means to the LSI and trained it to find and create 10 different clusters which can be seen below. In this case it can be hard to deduce how the different words cluster, because of the well mixed datapoints. It's though possible to identify some distinct clusters outside the big mixture of datapoints. The denser the clusters are gathered together the more similar context they share.

Subconclusion on patterns:

In our case the LDA-model was easier to interpret visual wise unlike the LSI, and it gave us some overall ideas about the topics and similarities. Overall the results from both models gives us an idea what this book is about, and who plays the biggest part of the in this case Killinger and Rotella. Furthermore we also identify that loans more precise subprime loans is the reason for the bank declining finical performance.

Predicting outcome of literature reviews of banking text:

In order to get estimate whether the literature, that this project is based on, is deemed good or bad, we scraped a number of reviews from Amazon, cleaned out the HTML and processed the data into a dataframe. We based a positive or negative review on the point given by the reviewer on a 5 point scale. 1-3 point was deemed negative. 4 or 5 point was deemed positive. Unfortunately, only 110 reviews were available. This is a very small dataset, but this time the real world limits us. A large part of the reviews were positive.

<u>SML approach using classifiers and oversampling.</u>

As mentioned, we only have 110 reviews. In addition, the positive and the negative reviews are imbalanced, heavily favouring the positive reviews. Our intention was to oversample the reviews and then undersample based on the minority class (the negative reviews). However, we discovered that this approach would create to much noise in the dataset and that we would have to clean it using regularization techniques such as Lasso or Ridge. It didn´t make much sense, and in the end we split the data in test and train sets and upsampled the train data using a resampling strategy based in only upsampling the minority class (the negative reviews). This only increased the number of reviews by 64, but we deemed it to aggressive to upsample any further.

We ran logistic regression on both with unsampled and upsampled traindata. The testscores were 85% with unsamples data and 64% with upsampled data. We fitted other models to the data, such as XGBoost and got the same result. This is not a satisfying result. We can deal with that in several ways. We could tune our models hyperparameters, sample the data differently (we did that) or gather more data.

The subconclusion here is, that we need more data in order for us to improve the model. That is out of the scope of this project, but more data could be gathered by scraping reviews form similar books on amazon.com or goodreads.com. We made an effort to resample the dataset without getting better results. However, we take responsibility for the method used and the pipeline that was build.

Countbased approcach based on positive/negative list.

The alternative to our logistic regression is a simple count based approach where the cooccurrence of words are counted. We ran a simple model counting positive and negative words in the reviews based in premade negative/positive word. The result was:

	Based on Amazon scores.	Count-based approach.
Total reviews	110	110
Negative reviews	13	69
Positive reviews	97	41

The result of the prediction was not very good. **The subconclusion** is, that this can be attributed to the fact that our positive/negative lists are based on material, that did not intend to review literature. Perhaps better lists with the right literature will improve the predictions. Also, we have no sematic meaning in the approach. An example of a positive review, that would be rated negative could be: “The people in the bank was assholes, dimwits and corrupt dryfood eaters. But the book was good.” Obviously, we would get a negative prediction using negative/positive lists.