Natural Language Processing: Document clustering & Topic Modelling

Sonja Tilly

January 2018

Project

- Identify the underlying structure of documents in an informative and intuitive way
- Cluster and visualise investment outlook documents
- Find key topics for each cluster

Approach & methodology

1) Data exploration
2) Pre-processing
3) Clustering
4) Results
5) Topic modelling
6) Conclusion
7) Criticism & ideas
8) Appendix

Data exploration

- 2018 economic outlook documents produced by 61 different asset management houses

Pre-processing

- Tidy up text
  - Convert all text into lower case
  - Remove page breaks
- Tokenize text
  - **Tokenization** is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens
- Extract nouns
  - Found to provide more meaningful output
- Lemmatize text
  - **Lemmatization** removes inflectional endings only and returns the base or dictionary form of a word ('lemma')
- Remove stopwords
  - Words that do not contain meaning

CLUSTERING

- Vectorize text
  - Convert a collection of raw documents to a matrix of TF-IDF features.

- Apply multi-dimensional scaling to convert vectors into 2 dimensions for plotting
  - Place each object in N-dimensional space such that the between-object distances are preserved as well as possible.
  - Each object is the assigned coordinates in each of the N dimensions.

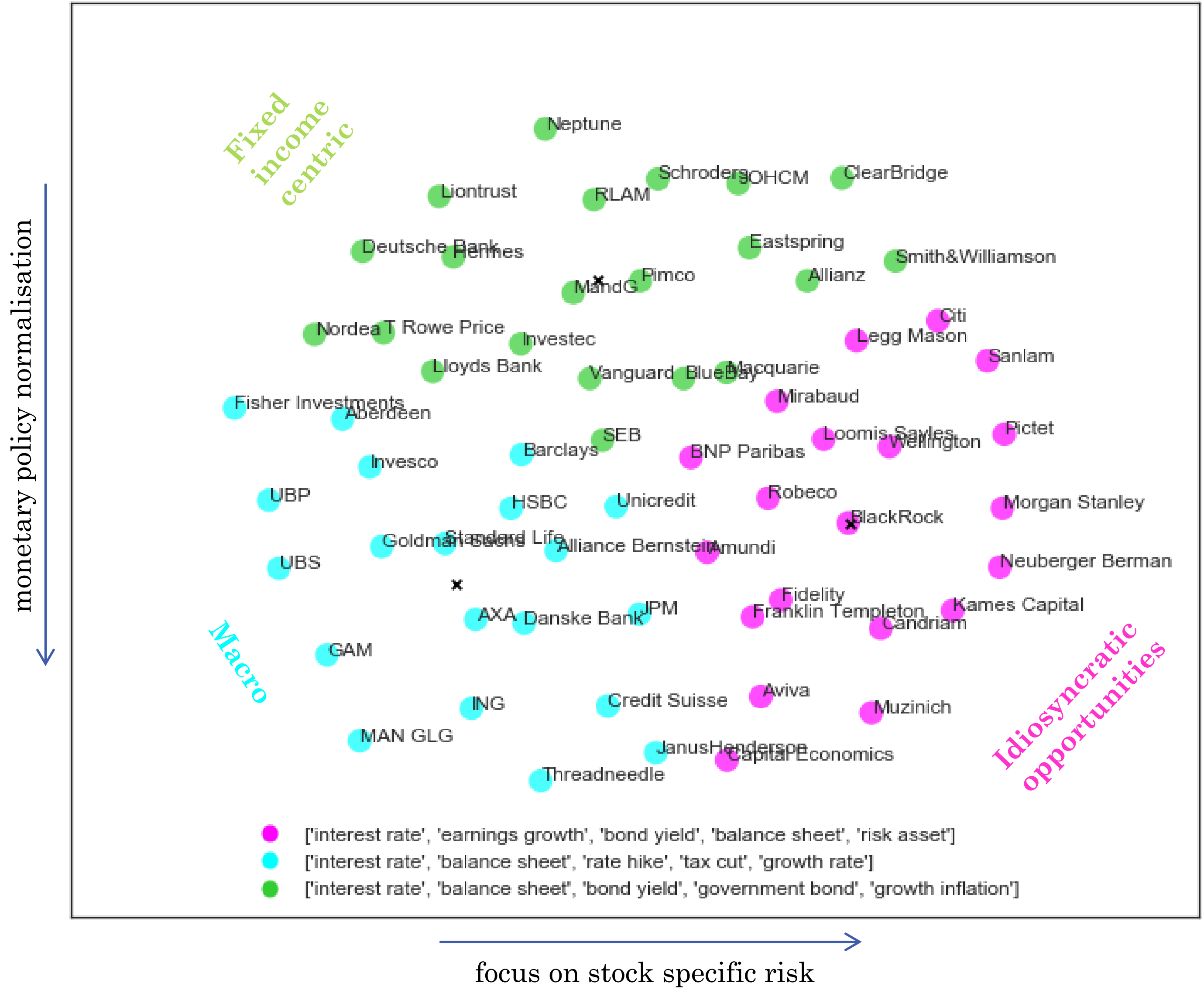
- Apply kmeans algorithm
  - Optimize number of clusters with silhouette score

Topic modelling – word frequencies

- Group document texts by cluster
- Vectorize text in each cluster
- Find most frequent bi-grams for each cluster

Note: bi-gram = combination of two words

Results: top bi-grams per cluster



Topic modelling – Latent Dirichlet Allocation

- Latent Dirichlet allocation (LDA) is a technique that discovers topics in a collection of documents
- Look up topic for each cluster
  - **magenta**: growth rate risk inflation equity bond policy asset credit yield
  - **cyan**: growth rate price risk bond equity sector term yield policy
  - **limegreen**: growth rate inflation price policy risk q term wage interest
- The results are very similar to the output from the kmeans cluster analysis

Conclusion

- Documents are very homogenous
- All clusters unanimously mention growth as most frequent word
- All clusters are concerned with interest rates
- Differences are nuanced
  - Policy normalisation vs. stock specific considerations
  - The three clusters can be described as focusing on fixed-income centric concerns, macro-economic perspectives and idiosyncratic opportunities.
- LDA topic modelling confirms findings from kmeans cluster analysis

Criticisms & ideas

- **Criticism**: Homogeneity of document content not ideal – little differentiation in views
- **Idea**: Use for monthly commentaries for one manager over a number of years and see how views (i.e. topics) evolved
- **Idea**: Use for monthly commentaries for all managers for one month and see how views (i.e. topics) differ
- **Idea**: Extract news article from news websites for topic modelling and sentiment analysis

Appendix I: TFIDF, Silhouette score

In information retrieval, **tf–idf** or **TFIDF**, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.
TD = frequency of word in a document.
IDF = measure of how significant a term is throughout the entire corpus (take log)
Score = TF*IDF

Source: https://en.wikipedia.org/wiki/Tf–idf

The **Silhouette Coefficient** is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The Silhouette Coefficient for a sample is (b - a) / max(a, b). To clarify, b is the distance between a sample and the nearest cluster that the sample is not a part of. Note that Silhouette Coefficient is only defined if number of labels is 2 <= n_labels <= n_samples - 1. The best value is 1 and the worst value is -1. Values near 0 indicate overlapping clusters. Negative values generally indicate that a sample has been assigned to the wrong cluster, as a different cluster is more similar.

Source: http://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

Appendix II: MDS

Multidimensional scaling (**MDS**) is a means of visualizing the level of similarity of individual cases of a dataset. It refers to a set of related ordination techniques used in information visualization, in particular to display the information contained in a distance matrix. It is a form of non-linear dimensionality reduction.

An MDS algorithm aims to place each object in N-dimensional space such that the between-object distances are preserved as well as possible. Each object is then assigned coordinates in each of the N dimensions. The number of dimensions of an MDS plot N can exceed 2 and is specified a priori. Choosing N=2 optimizes the object locations for a two-dimensional scatterplot.

The coordinates can be derived as follows: square each value in the distance matrix, double centre that such that the columns and rows both have a zero mean, and then take the singular-value decomposition (SVD) of that matrix. The point coordinates are then in the factors returned by the SVD.

A full derivation of this algorithm can be found here .

Source: https://www.revolvy.com/main/index.php?s=Multidimensional_scaling

Appendix III: LDA

Latent Dirichlet allocation (LDA) is a probabilistic topic model that assumes documents are a mixture of topics and that each word in the document is attributable to the document's topics.
LDA defines each topic as a bag of words, and you have to label the topics as you deem fit.
There are 2 benefits from LDA defining topics on a word-level:
1) We can infer the content spread of each sentence by a word count:
Sentence 1: 100% Topic F
Sentence 2: 100% Topic P
Sentence 3: 33% Topic P and 67% Topic F
2) We can derive the proportions that each word constitutes in given topics. For example, Topic F might comprise words in the following proportions: 40% eat, 40% fish, 20% vegetables, …
LDA achieves the above results in 3 steps.
<u>Step 1</u>
You tell the algorithm how many topics you think there are.
<u>Step 2</u>
The algorithm will assign every word to a temporary topic.
<u>Step 3 (iterative)</u>
The algorithm will check and update topic assignments, looping through each word in every document. For each word, its topic assignment is updated based on two criteria:
How prevalent is that word across topics?
How prevalent are topics in the document?

Source: algobeans.com