Unsupervised Document Clustering with Cluster Topic Identification

Michael Snow, Data Scientist

Big Data Team

Abstract

An important research goal at the Office for National Statistics (ONS) is to investigate the utilisation of textual data to gain insights from businesses. Motivated by this goal, this paper outlines methods for using document vectorisation, dimensionality reduction, clustering and singular value decomposition to gain insight from text documents, and demonstrates the use of these methods on free-text data about businesses. These methods will be useful in any context when processing large amounts of documents and wish to discover relationships and patterns in the data.

Contents

1 Introduction

This piece of work is inspired by a project the ONS's big data team have been exploring with data extracted from Companies House. Medium- to large-sized businesses are required to submit a full accounts document to Companies House and within this document the businesses typically include a description of the function of the company. This is potentially useful information to help classify the business into its relevant Standard Industrial Classification (SIC) code. This can also give insight into emerging new topics in industrial sectors; for example technology is moving at an incredible pace and there may not be sufficient granularity in the SIC classification to capture this. A recommendation of the Bean Review of Economic statistics [1] is that 'ONS should be proactive in pressing the case for the next industrial classification system to provide a richer picture of services activity', and utilising text data about businesses to produce new business classifications would help provide evidence for the need for this extra granularity.

The aim of the research described in this paper is to explore methods for using textual data about businesses - described in section 2 - to automatically generate a sentence describing the business to help inform business classification. This is challenging for a number of reasons; in particular, we are using 'strings' of information rather than numerical features. The initial aim is then to convert the strings of information into some numerical features, which we call word or document vectorisation. A naive approach to do this is to embed the documents based on their word count. For example, if we have a small dictionary set of words

$$D = \{\text{'I'}, \text{'do'}, \text{'not'}, \text{'like'}, \text{'to'}, \text{'cake'}, \text{'eat'}, \text{'no'}\}, \tag{1}$$

we could represent the string 'I like to eat cake' by the following row vector

$$V = [1, 0, 0, 1, 1, 1, 1, 0], \tag{2}$$

where the vector contains a '1' if the string contains the word in the respective dictionary position or a '0' if not. This naive approach could work well in some situations, for instance just discovering similarities based on the frequency of certain words, but a more sophisticated approach will also incorporate information from neighbouring words to gain more insight. To do this, we will use an algorithm called Doc2Vec, outlined in section 3.2.

2 Data

The data we have collected is from the Companies House website [2]. Each medium- or large-sized business upload yearly full accounts information for their company, which typically includes a sentence or small paragraph that describes the business' principal activities in that period. Below we show some typical examples of a company's self-description:The data we have collected is from the Companies House website [2]. Each medium- or large-sized business upload yearly full accounts information for their company, which typically includes a sentence or small paragraph that describes the business' principal activities in that period. Below we show some typical examples of a company's self-description:

- The company's principal activity is the manufacture and sale of wash-room systems.
- The company's principal activity continued to be management lease and charter of maritime vessels together with related marine services.
- The company's principal activity during the year continued to be the design and manufacture of heat exchangers and cooling equipment.

The data is uploaded in the form of a PDF which contains scanned images of the full accounts. The relevant information is able to be extracted using a combination of basic image processing, optical character recognition (OCR) to extract the text and regular expression (regex) to extract the relevant information. Our dataset consists of approximately 87,000 paragraphs of data similar to the examples above.

The main aim of the work conducted here is discover some common patterns in the data so we can gain new insights into SIC classification and potentially in the future construct a machine learning algorithm that automatically classifies a business based on its description.

3 Methodology

There are three main steps in this methodology after preprocessing the data. First we wish to vectorise our documents into useful numerical features. This embeds the documents into a vector space in $\mathbb{R}^d$ and the document embeddings are presented by a $n \times d$ matrix, where n is the number of documents. Next we wish to do some dimensionality reduction on the data to visualize potential patterns and trends in the data. The hope is that when we visualise the documents in two dimensions, we will be able to see clusters of documents with similar themes. An automatic

unsupervised clustering algorithm can then be performed on the data in two dimensions and we investigate whether the results are meaningful. Finally, we perform one last dimensionality reduction, using singular value decomposition on each of the clusters. The aim here is get a rank 1 (vector) approximation of each cluster's document matrix. This vector can then be used with the Doc2Vec model to infer a cluster's topic. This and each of these steps are explained in more detail in the proceeding methodology sections.

3.1 Data Preprocessing

Before considering trying to use the Doc2Vec algorithm described in the proceeding section, first the data has to be cleaned to produce more meaningful results. Many of the words in each document can be considered noise when building a model; for instance most of the descriptions contain words such as 'principal', 'activity', 'company'; as well as common words such as 'and', 'to' and so on. These words will contribute very little to the model and so we will remove them. We also remove things such as punctuation and digits as these also provide little information to discriminate business activity.

3.2 Document Vectorisation Using Doc2Vec

The Doc2Vec algorithm is an extension of the Word2Vec algorithm first proposed by Mikolov et al. [3]. The algorithm uses a shallow two-layer neural network approach to learn high-quality embeddings of words into a vector space. There are two approaches with Word2Vec, either a continuous bag of words (CBOW) approach or a skip-gram approach. We will be using the latter. With skip-gram, the input is a vector representation of a word v_{w_I} and the output is a context word w_O. The number of context words to the left or right of the input word is defined by a 'window' hyper-parameter. The objective function of Word2Vec aims to maximise the log probability of a context word w_O given its input word w_I, hence $\log P(w_O|w_I)$. Mikolov et al. discovered that negative sampling enabled accurate representations, especially for frequent words. They replaced the objective function by

$$\log \sigma \left({v'_{w_O}}^{\top} v_{w_I} \right) + \sum_{i=1}^{k} \mathbb{E}_{w_i} \sim P_n(w) \left[\log \sigma \left(-{v'_{w_i}}^{\top} v_{wI} \right) \right] \tag{3}$$

where σ is the sigmoid function in the skip-gram model. To classify documents, Mikolov extended

the Word2Vec architecture by taking the input to the model to be a special token of the document [4]. This is now commonly known as Doc2Vec (or sometimes Paragraph2Vec).

Using this approach we can embed all of our documents into a vector space; typically with a dimensionality of between $300 - 1000$. We can fine tune the hyper-parameters such as the learning rate of the neural network and the dimensionality to refine our model for more meaningful results. An interesting feature of using this model is that a new string can be given as an input to a pre-trained model and the model will infer a new vector into the vector space created by the model. It is then possible to see the nearest vectors under the cosine distance to the newly created vector; thus we can see the closest - or most similar - documents to the newly inferred string.

3.3 Dimensionality Reduction using t-Distributed Stochastic Neighbourhood Embedding

The t-Distributed Stochastic Neighbour Embedding (t-SNE) proposed by van der Maarten and Hinton [5], seeks a two- or three-dimensional embedding of the original high dimensional data for the purpose of visualisation. Here we focus on a two-dimensional visualisation of the data. t-SNE converts the Euclidean distance - or some defined metric - between two points into the probability of these two points being neighbours. The objective function of t-SNE which needs to be minimized is the Kullback-Leibler divergence between the probabilities in original space and mapped space. The objective function is given by

$$\Im(Y) = D_{KL}\left(P||Q\right) = \sum_{i \neq j} P_{ij} \log \frac{P_{ij}}{Q_{ij}}, \tag{4}$$

where P_{ij} are similarities of points x_i and x_j in the original space. It is typically given by

$$P_{ij} = \frac{\exp\left(-\dfrac{||x_i - x_j||^2}{2\sigma}\right)}{\sum_{k \neq l} \exp\left(-\dfrac{||x_l - x_k||^2}{2\sigma}\right)} \tag{5}$$

where σ^2 is the variance of the Gaussian that is centred on data point x_i. The t-SNE algorithm approaches the crowding problem by using the Student t-distribution with a single degree of freedom to model the similarities in the mapped space

$$Q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}}, \qquad Q_{ii} = 0, \tag{6}$$

where y_i are 2-D mapping of x_i.

Given the gradient of the t-SNE objective

$$\frac{\partial J}{\partial y_i} = 4X \sum_j (P_{ij} - Q_{ij})\,(y_i - y_j)(1 + ||y_i - y_j||^2)^{-1} \tag{7}$$

the t-SNE optimisation iteratively applies the update rule

$$Y^k = Y^{k-1} + \eta \frac{\partial J}{\partial Y} + \alpha(k)\left(Y^{k-1} - Y^{k-2}\right), \tag{8}$$

where $Y = [y_1, y_2, \cdots, y_N]$, η is the learning rate, and $\alpha(t)$ is the amount of momentum at the k-th iteration.

This approach to dimensionality reduction gives some truly excellent results. For example, on the well-known MNIST optical character recognition dataset shown in Figure 1; t-SNE is already able to cluster the digits very well in two dimensions.

Figure 1: Examples of MNIST digits.

An example of this is shown in Figure 2 taken from a tutorial by Metz [6]. The algorithm shows the natural clustering of the images based on their digit. Considering that each image has dimensions of 28×28, the algorithm considers this to be a 28^2- dimensional vector; so the result is visually very good and corresponds well to how humans visualise the difference in these images.

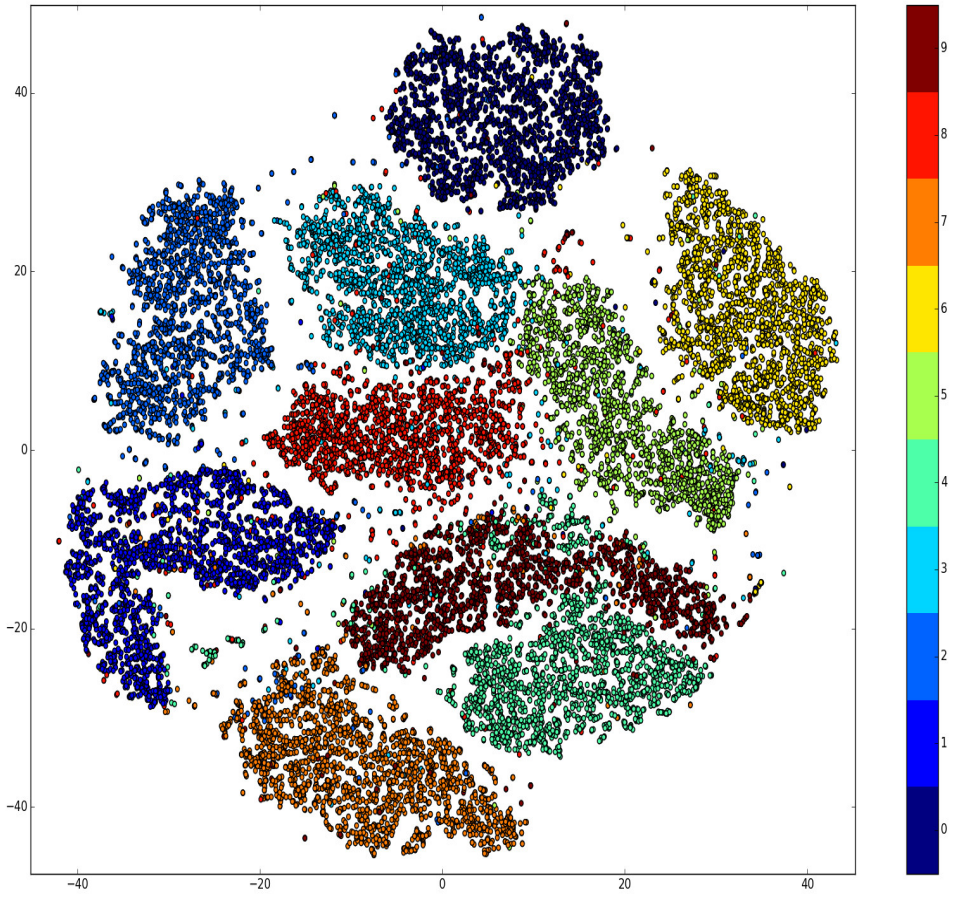

Figure 2: Example of t-SNE applied to the MNIST dataset.

3.4 Clustering using HDBSCAN

From our dataset, the expectation is that there will be lots of small clusters of business descriptions where similar business types have described themselves in a similar way. To effectively capture these clusters, we wish to identify the clusters based on the localised density of points in the vector space, rather than a less refined nearest neighbour method such as k-means.

HDBSCAN is a hierarchical, density-based clustering algorithm developed by Campello et al. [7] which builds on DBSCAN from Ester et al. [8]. DBSCAN sorts three types of points: core points on the interior of a cluster; noise on the exterior of a cluster and border points on the boundary of a cluster. If we let $D \in \mathbb{R}^d$, then a point is distinguished by the parameters ϵ and $MinPts$. If we define $N_\epsilon(p, D) := \{q \in D : |pq| \leq \epsilon\}$ to be a neighbourhood of a point p. A point p is a core point

if $|N_\epsilon(p, D)| > MinPts$, and a non-core point q is a border point if it is in the neighbourhood of a core point; else it is noise. Ester et al. define the DBSCAN clusters based on the idea of density-reachability which is detailed in their paper [8].

The advantage of using this algorithm is that it is very adept at accurately picking out clusters based on their density and connectedness rather than a nearest neighbour approach such as $k-$means clustering. This is demonstrated in Figure 3, with each cluster colour-code and noise represented as grey.

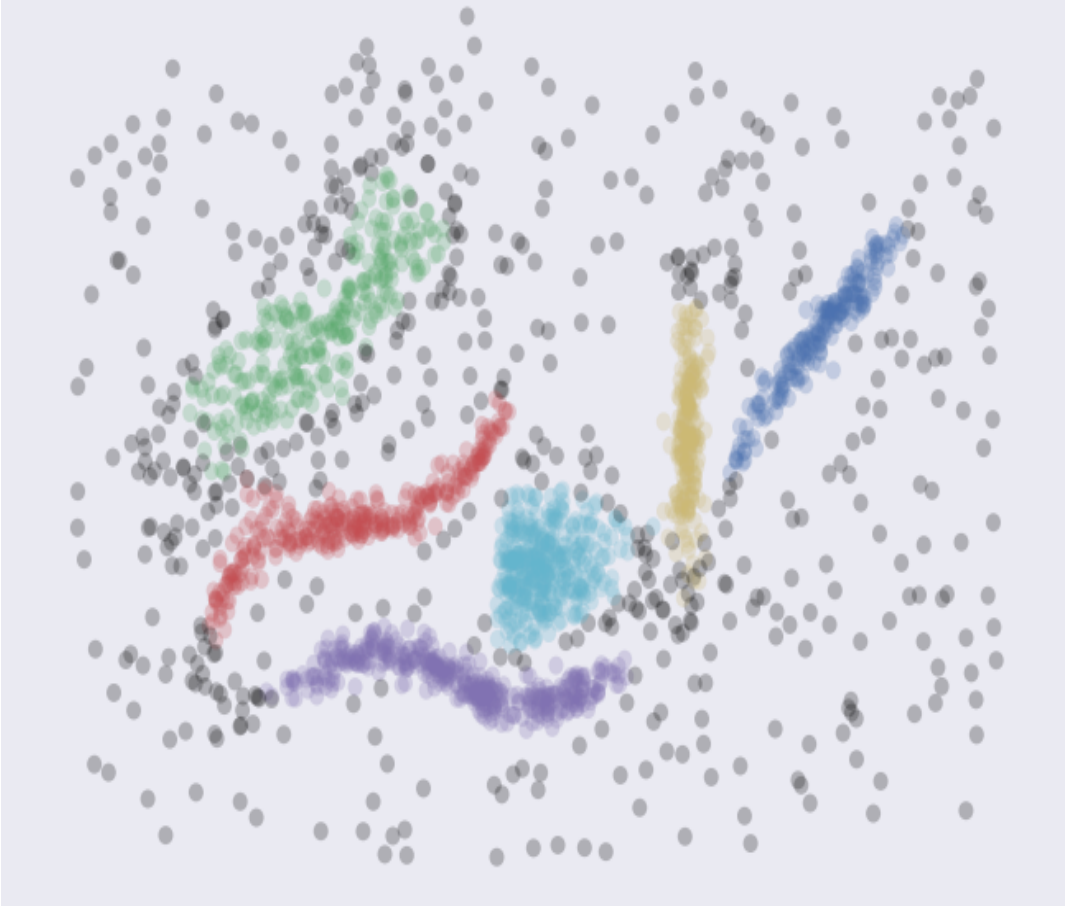

Figure 3: Example of HDBSCAN clustering

3.5 Cluster topic extraction using singular value decomposition

The final stage of the pipeline is to make an attempt to identify a general topic that represents each of the clusters. If the clusters are very representative of the data within the cluster and not too noisy, then one possible way to extract a description of the matrix would be to find a low rank approximation of the matrix. If a rank 1 vector representation of the matrix can be found, the resulting vector can then be inferred back into the Doc2Vec model and we can imply a cluster

topic based on the nearest document vector in the model to the inferred vector under the cosine distance.

To do this, a singular value decomposition (SVD) of the matrix can be found. The SVD is arguably the most well-known result from linear algebra and is a very useful matrix decomposition for a number of reasons; in this case because it provides the best low rank approximation of a matrix under the Frobenius norm.

Let M be an $m \times n$ matrix, whose entries are real (although they can also be complex). Then there exists a singular value decomposition of M such that

$$M = U\Sigma V^T, \tag{9}$$

where U is an $m \times m$ orthogonal matrix, V is an $n \times n$ orthogonal matrix and Σ is an $m \times n$ diagonal matrix. The columns of U and V are known as the left and right singular vectors respectively and the diagonal entries of Σ are ordered such that

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & 0 \\ & & \ddots & \\ & & & \sigma_{n-1} \\ & 0 & & & \sigma_n \end{bmatrix}, \tag{10}$$

and $\sigma_1 \geq \sigma_2 \cdots \geq \sigma_n$. These are known as the singular values of M, where their roots are the eigenvalues of M. If we denote $U = [u_1 u_2 ... u_m]$, $V = [v_1 v_2 ... v_n]$, then we can see that the SVD decomposes M into the sum of rank 1 matrices weighted by its singular values such that

$$M = \sum_{i=1}^{n} u_i \sigma_i v_i^T. \tag{11}$$

In this case we can construct a $k-$rank approximation of the matrix M by taking only the first k singular values of the matrix. In this case we have the k-rank approximation

$$\hat{M}_k = \sum_{i=1}^{k} u_i \sigma_i v_i^T, \tag{12}$$

where $\hat{M}_k$ is the *best* k-rank approximation to M under the Frobenius norm.

4 Proof of Concept Results

First a Doc2Vec model was created using the full dataset of nearly 87,000 documents; i.e. paragraphs describing businesses as described in section 1. This embedded each document into a 500-dimensional vector space. The research by Mikolov et al. [3] suggests that when training the model a dimensionality of between 300 and 1000 dimensions works well. To give insight into how well the model has been clustering, some document vectors were inferred into the model and the nearest vectors (by cosine distance) in the model were outputted. If the model is providing meaningful results, we expect the nearest vectors to be similar to each other and the inferred vector - i.e. there are clusters around the inferred vector.

To test this, we tested the string 'elderly disabled care'; here Doc2Vec infers a vector and we can check the nearest documents to the vector. The results were as follows:

- The principal activity of the company continued to be that of the provision of long term care to the elderly and young disabled.
- The principal activity of the company during the year was the provision of residential care services for the elderly and disabled.
- The principal activity of the company during the year was the operation of a hotel for the disabled.
- The principal activity of the company during the year was residential care for the elderly and disabled.
- The company's principal activity during the year was that of sales and servicing of the equipment for the disabled.

This shows the model is working well; the inferred vector is close to a group of documents that are represented very well by the phrase 'elderly disabled care'. This also suggests the model has

clustered similar documents very well too. As well as being quality-assurance, this method of inferring vectors into the model may be useful in and of itself; for example, to find businesses which are engaged in a particular economic activity of interest. It should be noted that if a new string contains a vocabulary that is completely disjoint from the dictionary the model was trained on, the model will not know what to do as it has not seen that information. In this case it initializes a random vector and so the result would be meaningless.

We can now visualise this using t-SNE as described in section 3.3.

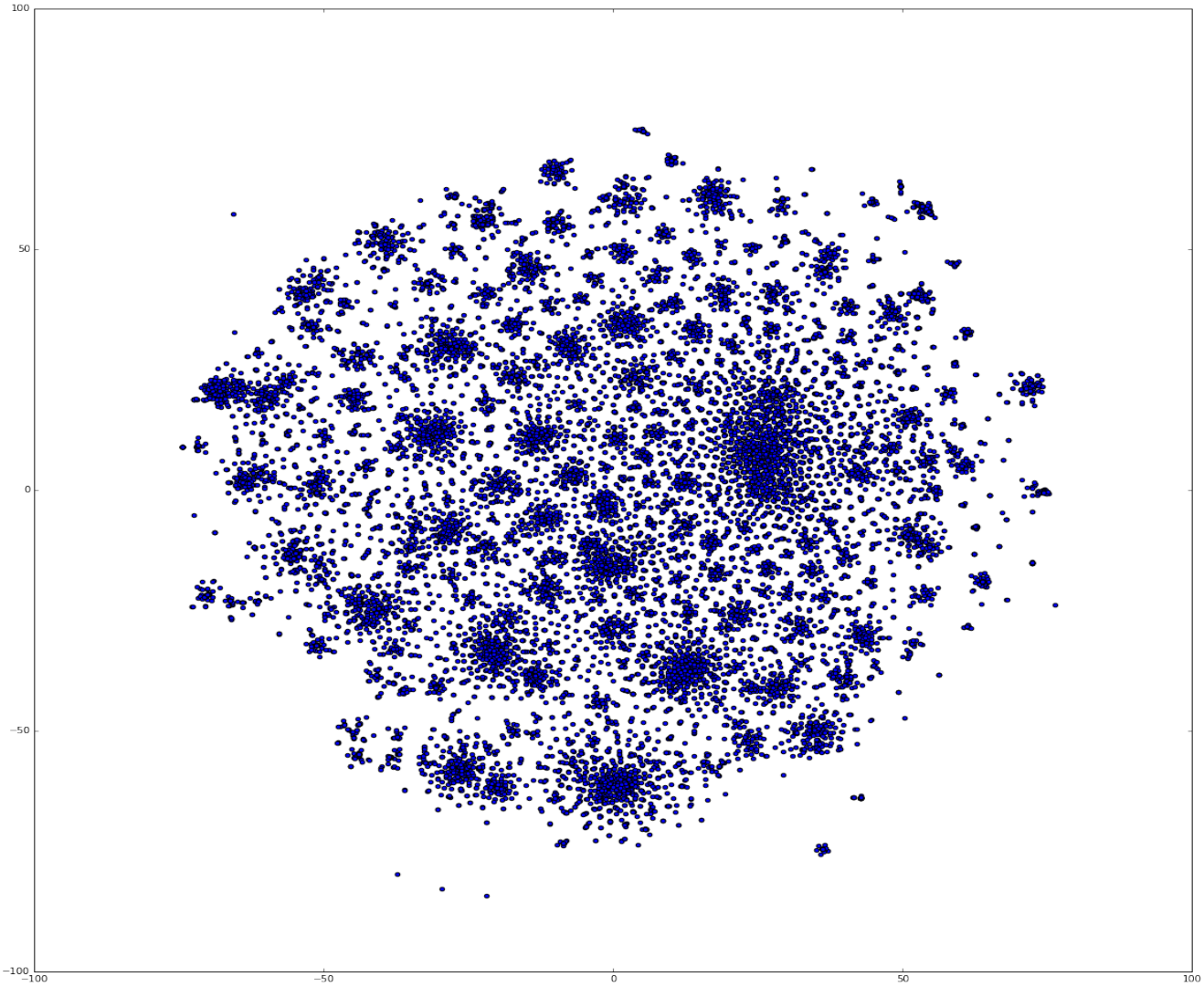

Figure 4: Dimensionality reduction using t-SNE

Figure 4 illustrates the vector space in two dimensions after dimensionality reduction. We can now see there seems to be many small clusters of documents grouped together. We cluster this based on their densities using HDBSCAN, described in section 3.4.

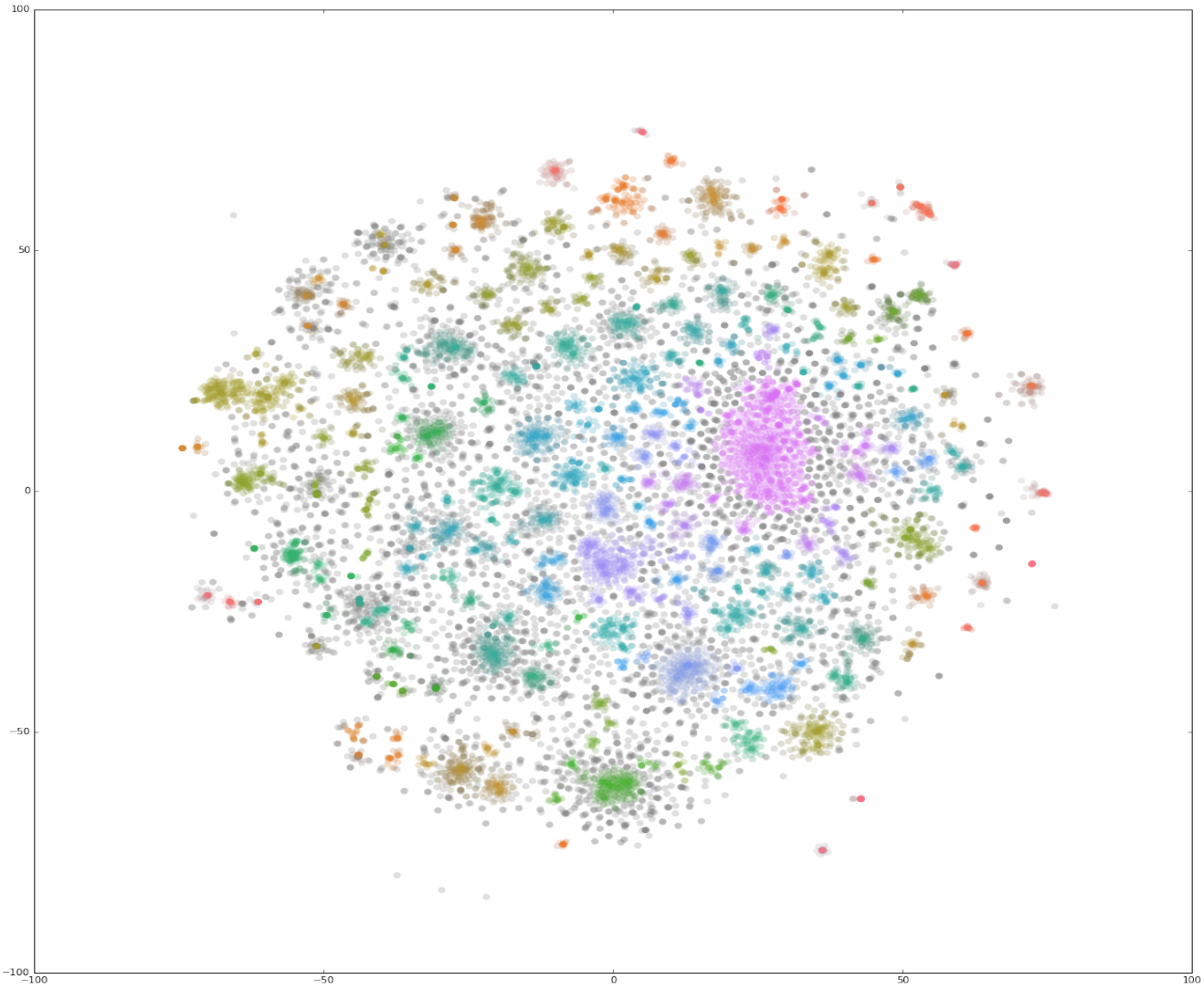

Figure 5: Density-based clustering using HDBSCAN

Figure 5 colour codes the different recognised clusters. In this case, although it is hard to visualise such a spectrum, HDBSCAN identified over 4000 separate clusters containing at least 2 businesses. If we investigate business descriptions around these clusters we get some interesting results. These are examples of 3 clusters and business descriptions within those clusters.

Cluster 1:

- Principal activities. The company is a wholly owned subsidiary of a group whose principal activities are that of development and management of solar farms.
- Principal activities. The company is a wholly owned subsidiary of a group whose principal activities are that of development and management of solar farms.

- Principal activities in the year under review is that of the development, construction and operation or sale of wind farms; provision of commercial asset management services to wind and solar farms and provision of operational management services to wind farms.

Cluster 2:

- The principal activity the principal activity of the company is to provide student residential accommodation.
- The principal activity the principal activity of the company is to provide student residential accommodation.
- The principal activity the principal activity of the company is to provide student residential accommodation.

Cluster 3:

- The company's principal activity is providing data storage services and solutions.
- The company's principal activity is providing data storage services and solutions.
- The principal activity of the company was that of an independent integrator of data storage and data management solutions.

As we can see, through exploration of the clusters, different business classification topics can clearly be identified. This will not always be the case; for instance some of the larger clusters will contain noise where the model is picking up certain phrases which are not related to the business activity. These examples also pick out some potentially interesting features to inform SIC classification. With the recent explosion in big data, cluster 3 shows a cluster containing businesses selling data storage and management solutions; a topic not captured in the previous SIC update. Cluster 2 shows an emerging area in the private lettings of student accommodation. Interestingly in this cluster, approximately 75% of the businesses are correctly labelled under the SIC class of 'Real estate activities', while approximately 25% of the companies are labelled under 'Accommodation and food service activities'. This class is aimed at things such as hotel stays and the hospitality industry, suggesting these businesses may be classified to an incorrect SIC group. The model also picks out established industries very well, such as banking, property or for example in cluster 1 the production and maintenance of solar farms.

With large amounts of data and clusters, generally we will not want to manually give each cluster a topic. If we assume that the data is very similar within clusters (i.e. we assume that the Doc2Vec model has produced a meaningful transformation), this justifies the use of singular value decomposition. Here we collect a separate matrix containing the document vectors for each separate cluster. We then take an SVD of each cluster's document matrix and return a rank 1 vector representation of the matrix. The vector can then be compared via the cosine distance to all other documents in the Doc2Vec original vector space, and the closest vector is returned. This document can be used to describe the cluster as a whole, or a topic can be manually inferred from the document description. The technique can be used to automate cluster topic identification, especially when - like in this case - there are thousands of clusters and manual labelling is prohibitive.

The results for the three example clusters shown earlier are as follows:

Cluster 1:

- **Cluster topic**: Principal activities. The company is a wholly owned subsidiary of a group whose principal activities are that of development and management of solar farms.

Cluster 2:

- **Cluster topic**: The principal activity the principal activity of the company is to provide student residential accommodation.

Cluster 3:

- **Cluster topic**: The company's principal activity is providing data storage services and solutions.

The topic descriptions are each very representative of the clusters shown as they each relate to a business within the cluster. From this process, it has been shown that using the proposed machine learning pipeline, an automated, unsupervised machine learning process can be used to cluster documents and also infer a cluster topic.

5 Discussion and Further Work

From the results we can see that this methodology of creating a document vector space using Doc2Vec, followed by dimensionality reduction, clustering, and singular value decomposition produces meaningful results. In general, the final result will depend on a few different areas. First, we have trained the Doc2Vec model using a relatively small amount of data in big data terms. Although this seems to be sufficient to produce meaningful document embeddings, to produce results with more confidence a large training set would be needed. With an insufficient amount of data to train with and poor preprocessing of the documents, this may not produce a sufficient result. Parameters can also be tuned with both dimensionality reduction and clustering algorithms too; t-SNE can be tuned to focus on local or global information and similarly, HDBSCAN can be tuned to ignore small clusters. If a cluster is too noisy, then the singular value decomposition of the cluster's document matrix may not be sufficiently representative as it may pick out a noisy vector. This is a consequence of the model not being well trained rather than using the SVD approach.

In our application, although the model has produced meaningful embeddings, a lot of the clusters are dominated by popular business types, such as holding companies, financial services and property management and investment. To really capture some more granular business information, we conclude that more data is needed to train the model. Despite this, the model provided some interesting insights towards informing SIC. The model is capable of picking out some emerging trends and showing potential misclassification. It is also useful to explore business types by inferring a vector into the model; so for instance if we are interested in renewable energy, we can create a vector of some key words and discover businesses close to that description.

For future work we would like to explore this approach with a richer source of data. The pipeline can be further extended to generate an automatic classification system of businesses based on the document vector space embeddings. This vector space could be used as a feature space for machine learning classification.

6 Pipeline Jupyter Notebook

A Jupyter Notebook showing an example of this pipeline using the data extracted from Companies House is available at the following link:

- https://github.com/ONSBigData/Clustering_paper

References

[1] C. Bean, "Independent Review of UK Economic Statistics," *Office for National Statistics*, 2016.

[2] "Companies house archive." https://beta.companieshouse.gov.uk/. Accessed: 25/09/2017.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[4] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *CoRR*, vol. abs/1310.4546, 2013.

[5] L. van der Maaten and G. Hinton, "Visualising Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, 2013.

[6] "Visualising with t-sne." https://indico.io/blog/visualizing-with-t-sne/. Accessed: 25/10/2017.

[7] R. J. Campello, D. Moulavi, A. Zimek, and J. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 1, p. 5, 2015.

[8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *Proc. 2nd Int. Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 226–231, 1996.