The Alan Turing Institute

Data Study Group Final Report: Spend Network

8 – 12 April 2019

Automated matching of businesses to government contract opportunities

https://doi.org/10.5281/zenodo.3558243

This work was supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1

Spend Network

Luis Ospina-Forero
University of Manchester

Mihai Cucurigu
University of Oxford
The Alan Turing Institute

Oduwa Edo-Osagie*
University of East Anglia

Hoang Le*
Trinity College Dublin

Silviu Oprea*
University of Edinburgh

Bemsibom Toh*
Heriot-Watt University

Erin Clark*
Spend Network

Aude Vuilliomenet*
University College London

Kai Xu*
University of Edinburgh

Yuchen Zhao*
University of Southampton

Abstract

This report presents the output of a week-long collaboration between Spend Network, and lead academics from the University of Manchester and the University of Oxford that attended the Data Study Group at The Alan Turing Institute. Spend Network is a platform that aims to enable efficient public procurement. The goal of this work was to match suppliers to tenders. Our approach consists of building vector representations of suppliers and tenders and identifying their compatibility with the distance between the respective vectors. In building their representations, we make use of both their textual descriptions and the knowledge of previously awarded contracts. We find previous contracts informative of future procurement decisions. Our best results use Correlated Topic Models (Blei et al., 2007) for extracting representations of textual descriptions.

1 Introduction

Spend Network is a platform that aims to enable efficient public procurement, by connecting suppliers to government tenders. The goal of our project was to find discover connections between tenders posted on the platform and suppliers registered on the platform.

We had two applications in mind. First, to use these connections to improve relevancy of results when suppliers search for tenders. Second, to build a recommender system that would alert relevant suppliers when a new tender is released.

*All authors had an equal contribution.

1.1 Data Overview

We had access to historical data from Spend Network. The data contains information in terms of different aspects in public procurement, and is divided into two files.

The first file contains information about previous contracts. Once a supplier has won the bidding process to carry out goods, works, or services, detailed in a tender, and they have made a contract agreement with the buyer, a contract document is published. This shows a successful connection between a supplier and a tender.

The second file contains public raw text data scraped from each supplier's website.

1.2 Methodology Overview

We employed several methods of discovering connections between tenders and suppliers. For each tender and supplier, they all proceed by first building a tender representation from the textual description of the tender, a supplier representation from the textual description of the supplier in conjunction with text scraped from their website. Then, they assess the appropriateness of the supplier for the services demanded by the tender by looking at the similarity between the two representations.

The first category of approaches are completely unsupervised and only make use of the text data described above. Such approaches build the supplier and tender representations as low-dimensional embedding vectors of the corresponding texts. We call such an approach a *textual embedding approach*.

The second category uses not only the description text to build embeddings, but also exploits previous connections, that is, previous successful contracts, in our dataset. They embed information about the topology of the graph of previous connections in these embeddings. We call such an approach a *graph embedding approach*.

The third category only looks at text, but trains topic modelling systems on our dataset and represents each entity (supplier or tender) as their per-document topic proportions, where a document consists of the textual description. We call such an approach a *topic modelling approach*.

1.3 Main Conclusions and Limitations

Graph embedding approaches proved more beneficial and robust to irregularities than textual embedding approaches, particularly when one supplier seemed to dominate a lot of past connections. Topic modelling approaches proved most beneficial of all in recommending tenders to suppliers. All in all, our methods could be used to refine Spend Network's search and recommendation system.

Our data, especially the subset of it which was scraped from websites, is particularly noisy, especially in the sense of including information that would not be useful to particularise a specific supplier. This proved disruptive of the performance of our methods. The textual description, apart from website data, was also missing for some of our suppliers.

Furthermore, previous connections only included information about suppliers that won the bid for a tender, not about suppliers who bid and did not win. Such side information could provide important additional insights regarding the assessment of a match between a supplier and a tender.

Finally, there was no structured mapping from contract data to the set of tenders. This mapping had to be inferred based on the description of the associated tender attached to each contract.

2 Data Analysis

In this section we provide a more detailed description of our datasets, our data analysis, the preprocessing steps we employed.

2.1 Datasets

We have two datasets to discuss. The *contracts* dataset and the *supplier website text* dataset.

2.1.1 Contracts Dataset

The contract dataset discussed above contains 11,815 contract notice records with 1,288 unique government buyers and 3,592 unique suppliers. Each record in the dataset has 27 fields as follows:

- json: the format of the OCDS content, containing all information of the tenders;
- ocid: global unique ID of the notice, comprised of a prefix, a source id and a notice id;
- source: indicates the source the data was collected from;
- releasedate: date the information was first released (equal to, or before the publish date);
- aw_title: title of the award;
- aw_description: description of the award;
- cpv_code: category codes related to the tender notice;
- cpv_string: description of the primary CPV code;
- level1: government level one;
- level2: government level two;
- level3: government level three;
- buyer_id: official ID of the buyer;
- buyer: entity purchasing the goods or services;
- countryname: country of the buyer;
- supplier: the supplier the tender was awarded to;
- supplier_id: official ID of the supplier;
- supplier_type: company type of the Supplier;
- supplier_is_sme: whether the supplier is a small or medium enterprise;
- supplier_locality: locality of the supplier;
- supplier_postcode_geo_point: suppler geo postcode;
- supplier_status: status of supplier organisation;

- aw_contract_start: starting date of contract, where provided;
- aw_contract_end: ending date of contract, where provided;
- aw_value_string: total value of award, as a string;
- clean_aw_val: clean value as float, 0 if non numeric;
- aw_currency: currency of the value;
- text: concatenation of the title, description, aw title and aw description fields.

2.1.2 Supplier Website Text Dataset

The supplier website text dataset contains 255 records of the textual information from the websites of the suppliers. Each record has 4 fields as follows:

- company_name: name of the company;
- company_url: URL of the companys homepage;
- home_page_text: text from the companys homepage;
- about_or_contact_text: text from the companys *about us* or *contact us* page;

2.2 Exploratory Data Analysis

To explore the contract dataset, we start by visualising the bipartite graph between the government and the suppliers (4,880 nodes in total), which is shown in Figure 1.

In the figure, each entity (buyer or supplier) is marked with a circle, of which the radius is proportional to the number of contracts the respective entity is involved in. We can see that there are a few buyers which are very popular. Thus, it is interesting to identify which local subgraphs have a high proportion of edges present, i.e. are more dense. One way of assessing the density of node neighbourhoods is by computing the graph Laplacian, $L = D - A$, where A is the adjacent matrix of the graph and D is its diagonal degree matrix. We consider the eigenvector corresponding to the second largest eigenvalue, v_2 which can be subsequently used to give an indication of how dense the local neighbourhood of a node is. Here, we mark the top 20 buyer nodes according to v_2, and mark their neighbour suppliers (447 in total). They cover 5443 contracts in the data sets. The visualisation for this is shown in Figure 2.

Finally, another interesting fact about the data set is that there are actually two disjoint subgraphs. This is found by looking at the eigenvalues of the Laplacian corresponding to the adjacent matrix A, of which the number of leading 0s indicates the number of connected components. By looking at the actual subgraphs, we find that one of them contains 4,841 nodes with 1,275 buyers and 3,566 suppliers, and the other has 39 nodes with 13 buyers and 26 suppliers. It could be relevant to investigate why are these subgraphs disconnected from one another.

2.3 Data Preprocessing

We preprocess the data sets to extract the needed information to train our systems and to generate the data needed to evaluate the performance. As the ground truth (i.e., which contract was awarded to which supplier) is contained in the contract data set, we use it as a main source to extract the information about contracts and suppliers. We extract following fields:

- ocid (renamed as contract_id): the ID of the contract;
- supplier (renamed as company_name): the name of the supplier of the contract;
- aw_title + aw_description + cpv_string + level1 + level2 + level3 (concatenated and renamed as contract_text): the textual information of the contract, including its title, its description, the string that describes its procurement category, and three strings related to its government level.

We extracted the textual information of a company from the company's homepage. For each supplier, we use its name (company_name) as an index, based on which we look for the information of the supplier's website from the second data set, i.e. the supplier website text data set. If a record can be found, we use the text from the supplier's homepage (i.e., home_page_text, renamed to company_text) and the text from the supplier's *about us* or *contact us* page (i.e., about_or_contact_text, renamed to company_info).

The preprocessed data includes five fields (three from the contract data set and two from the supplier website text dataset. To generate the data

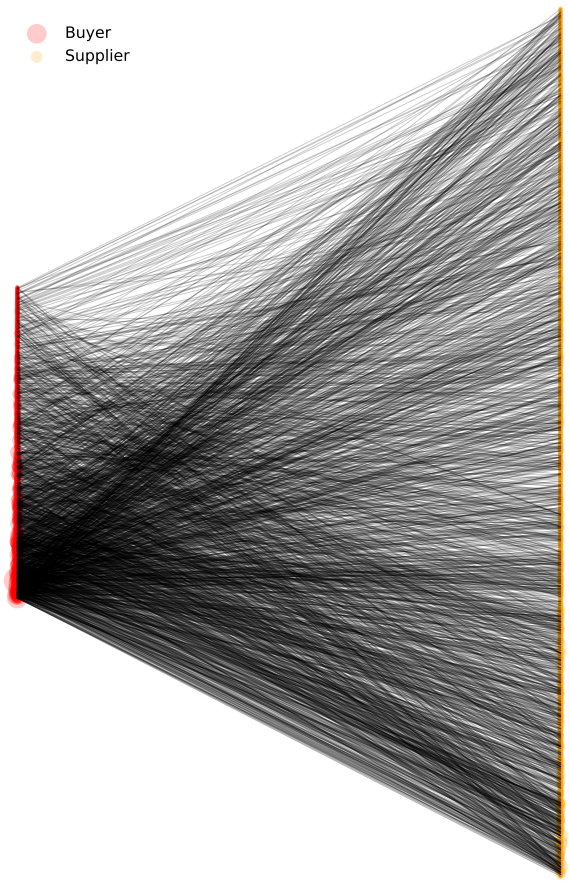

Figure 1: Links from the contract dataset; only a random 20% of the total amount of links are shown.

for training our systems and evaluating their performance, we randomly sample 90% of the preprocessed data as the training set and use the remaining 10% as the testing set.

When we generated companies' textual information from the website description data set, some companies' names from the contract data set could not be found. We only kept the companies with textual information in our training and testing sets. Our final training set had information about 13650 contracts, while the test set had information about 1740 contracts.

3 Methodology

In this section we formulate the computational problems we are addressing, introduce notational conventions, and suggest approaches for each problem. Let $S = \{s_1, s_2, \ldots\}$ be a set of suppliers and $T = \{t_1, t_2, \ldots\}$ be a set of tenders.

We define the SEARCH problem as follows. For a given supplier that enters the Spend Network platform, retrieve those tenders that are most compatible with the services that the supplier offers. Formally, let n be an integer. The purpose is to find a mapping $\sigma^n \subset S \times T^n$ with $\sigma^n(s_i) = t_{\pi_1^i} t_{\pi_2^i} \ldots t_{\pi_n^i}$ where $s_i \in S$, $t_i \in T$, and π^i is the utility ordering function of supplier s_i. That is, $t_{\pi_j^i}$ is the tender that has precedence j according to the preference of s_i.

We also define the RECOMMEND problem as follows. For a given tender posted on the Spend Network, retrieve those suppliers that are most suitable to the requirements of the tender. Formally, let n be an integer. The purpose is to find a mapping $\tau^n \subset T \times S^n$ with $\tau^n(t_i) = s_{\pi_1^i} s_{\pi_2^i} \ldots s_{\pi_n^i}$ where $t_i \in T$, $s_i \in S$, and π^i is the utility ordering function of tender t_i. That is, $s_{\pi_j^i}$ is the supplier that has precedence j according to their suitability to be awarded tender t_i.

We suggest the following approach for the problems defined above. First, create vector representations of both tenders and suppliers. We use the term *embedding* to refer to such a representation. Then, for SEARCH, given tender t_i, retrieve the top n suppliers from S according to the similarity between the embedding of each supplier and the embedding of t_i. That

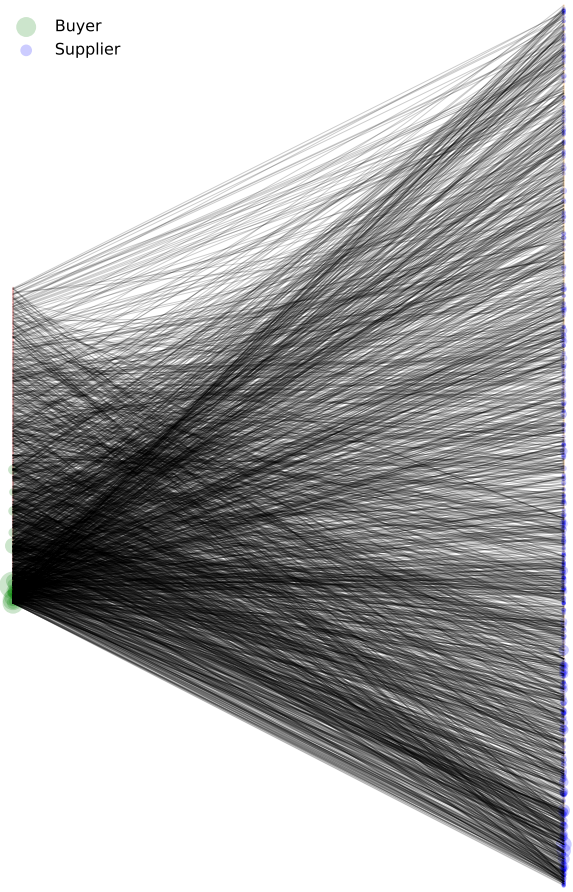

Figure 2: Same as Figure 1 with only top 20 most popular buyers marked with green circles and their suppliers marked in blue circles (447 in total).

is, $s_{\pi_1^i} = \arg\max_{s_j \in S} \cos(t_i, s_j)$ and so on. For RECOMMEND, given supplier s^i, we proceed in an analogous manner. That is, $t_{\pi_1^i} = \arg\max_{t_j \in T} \cos(s_i, t_j)$ and so on.

What makes our approaches different from each other is the way they create supplier and tender embeddings, and the similarity metric they use. This is what we focus on in the following sections.

3.1 Textual Embeddings

The first way to create tender and supplier embeddings that we suggest is completely unsupervised. We first extract textual descriptions of each supplier and each tender, as specified in Section 2. We then use the ParagraphVector model (Le and Mikolov, 2014) to create embeddings of each text. As the similarity metric we use cosine similarity between embedding vectors. We call this the TEXT approach.

3.2 Filtered Textual Embeddings

We proceed as specified in the previous section, but we filter out all words from the NLTK list of English stopwords (Bird, 2006) from each text before creating the embeddings. We call this the F-TEXT approach.

3.3 Graph Embeddings

The unsupervised approaches suggested so far have the clear advantage of being able to handle arbitrary tenders and suppliers, as long as these have an associated textual description that can be accessed. However, they make no use of historical interactional patterns between tenders and suppliers. In this section we include historical information in the tender and supplier embeddings that we build.

We proceed by representing tenders, suppliers, and previous interactions between them in a bipartite graph, as shown in Figure 3. Each node in the first partition represents a tender that has previously been assigned to suppliers, and each node in the second partition represents such a supplier. Each undirected edge between one suppliers and one tender indicates a previous assignment of that tender to that supplier. Note that a supplier can

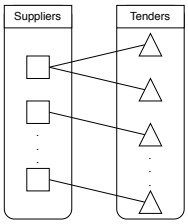

Figure 3: Example of a bipartite graph between connecting suppliers to tenders, as discussed in Section 3.3

satisfy multiple tenders. We assumed a tender can be assigned to exactly one supplier, which was the case in our training data, and is the most common real-life scenario. In building corresponding embeddings, we would like to exploit the topological structure of this graph. Intuitively, the nature of a supplier should be deduced not only from the textual description that the supplier provides about themselves, but also by looking at tenders that they were previously assigned. Furthermore, considering all previous tenders should provide more information than considering only one historical tender. Similarly, the nature of a tender should be deduced not only from its textual description, but also by looking at the supplier it was assigned to.

We now describe how we exploit the topology of such a graph to build embeddings for the tenders and suppliers represented in that graph. We use the Structural Deep Network Embedding method (SDNE) suggested by Wang et al. (2016). Intuitively, it builds an embedding of each node in a graph while trying to preserve first and second order proximity information. Two nodes are in first order proximity if they are connected by an edge. Two nodes are in second order proximity if they share many neighbouring nodes, i.e. there is a big overlap between their neighbourhoods. Formally, let $G = (V, E)$ be a weighted graph with nodes $V = \{v_i\}_{i=1}^{|V|}$ and edges $E = \{e_{i,j}\}_{i,j=1}^{|E|}$. Each edge $e_{i,j}$ connecting nodes v_i and v_j has an associated weight $s_{i,j} \geq 0$. $s_{i,j} = 0$ marks the fact that the two vertices are not connected. For any pair of vertices v_i and v_j, if $s_{i,j} > 0$ we say there exists first order proximity between the two vertices. Second order proximity, on the other hand, describes the proximity between the neighbourhoods of the two nodes. Let $\mathcal{P}_i = \{s_{i,j}\}_{j=1}^{|V|}$ denote the first order proximity between v_i and the other vertices. The second order proximity between v_i and v_j is specified by the similarity between $\mathcal{P}_i$ and $\mathcal{P}_j$, which indicates the number of common neighbours that v_i and v_j share. In accounting for second order proximity, we assume that two vertices are similar if they share common neighbours. This has been shown to be a reasonable assumptions in several fields. For instance, in linguistics, the distributional hypothesis assumes two words are are in semantic proximity if they tend to be surrounded by similar contexts (Dash, 2008). Social media users tend to be friends if they share many common friends (Jin et al., 2001).

The purpose of SDNE is to find a mapping $f : V \rightarrow \mathbb{R}^d$ where $d \ll N$. If $f(v_i) = \boldsymbol{y}_i$ and $f(v_j) = \boldsymbol{y}_j$, the objective is to make the similarity between $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ preserve first-order and second-order proximity between v_i and v_j. To account for first order proximity, the model uses a loss term which applies a penalty whenever similar nodes are mapped far away in the embedding space $\mathcal{R}^d$, referred to as first-order loss. This objective is similar to that employed in Laplacian Eigenmaps (Belkin and Niyogi, 2003). To account for second order proximity, the model passes the adjacency matrix of the graph through an autoencoder (Rumelhart and McClelland, 1987) and identifies the second-order loss with the reconstruction loss. The first and second order losses are jointly minimised. For implementation details we direct the reader to the original paper (Wang et al., 2016). In our experiments, we make use of their implementation published at `https://github.com/suanrong/SDNE`.

Using SDNE we construct node embeddings of our bipartite graph of tenders and suppliers. Each embedding preserves information about historical interactions between entities (tenders and suppliers). The SEARCH and RECOMMEND problems can be easily addressed using the familiar cosine similarity between corresponding embeddings if the entities of interest are present in the graph. However, we need an extra mechanism in place to deal with unseen entities. Specifically, we need two devices: (1) A way of finding existing correspondents to unseen tenders (unseen suppliers for RECOMMEND), in order to exploit historical interactional patterns exhibited in our graph between those correspondents and existing suppliers (existing tenders for RECOMMEND) and (2) An utility ordering function for ranking those existing suppliers (existing tenders for RECOMMEND).

We illustrate our suggested devices with the following example, depicted in Figure 4. Assume we

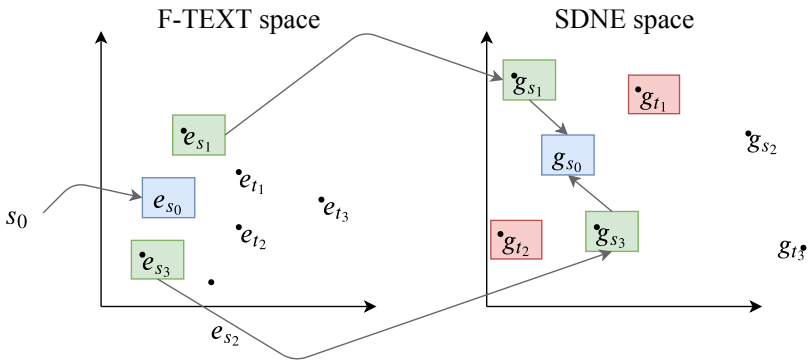

Figure 4: Mechanism for recommending tenders to an unseen supplier s_0, as discussed in Section 3.3. $s_{1:3}$ and $t_{1:3}$ are existing suppliers and tenders, respectively. g_{t_1} and g_{t_2} represent tenders of similar proximity to g_{s_0}, which we recommend according to the utility scheme discussed in Section 3.3.

have the set of suppliers $S = \{s_1, s_2, s_3\}$ and the set of tenders $T = \{t_1, t_2, t_3\}$. Let e_{s_i} and e_{t_i} be the F-TEXT embeddings of supplier $s_i \in S$ and tender $t_i \in T$, respectively. Similarly, let g_{s_i} and g_{t_i} be the SDNE embeddings of supplier $s_i \in S$ and tender $t_i \in T$, respectively. We address the SEARCH problem first. Specifically, let s_0 be an unseen supplier for which we would like to produce a sequence of recommended tenders. We proceed by computing the F-TEXT embedding e_{s_0} of s_0 based on its textual description. We then consider the closest suppliers of s_0 in the F-TEXT space. Assume these are s_1 and s_3. The embedding of s_0 in the SDNE space, g_{s_0}, is then obtained as the centre of mass of g_{s_1} and g_{s_3}. We recommend tenders in the order dictated by their closeness to g_{s_0} in the SDNE space. However, to order tenders of similar closeness, we implement a utility scheme. Specifically, to ensure equal opportunity, we perform a random uniform ordering. Note that alternative utility schemes are possible, depending on the objective. For instance, the tenders may be ordered by urgency, to ensure fast completion, or by price, to ensure maximum profit.

As the similarity metric, we use the cosine similarity between the SDNE embeddings.

3.4 Merged Embeddings

A further approach for building supplier and tender embeddings that we suggest is merging F-TEXT and SDNE embeddings by averaging. Our merging strategy was plain arithmetic mean. Future implementations should consider better merging strategies, that take into consideration the rotation and direction of vectors within the individual embedding spaces. We use the term MERGED to refer to the resulting embeddings. We also use cosine similarity as the embedding similarity metric.

3.5 Automated Matching by Correlated Topic Modelling

In the final approach that we suggest we use Correlated Topic Models (CTM) (Blei et al., 2007) to create topic representations of textual descriptions of suppliers and tenders.

CTM is a generative topic model. As such, it assumes that documents have been generated by sampling probability distributions. Its goal is to recover these distributions by using posterior inference. From a Machine Learning point of view, this is a case study in applying hierarchical Bayesian models to grouped data. The advantage of using a generative model is its natural ability to predict relevant matches for suppliers and tenders that do not appear in our training set. CTM is an extension of Latent Dirichlet Allocation (LDA). The idea behind LDA is to assume that each N-word document $\overline{w}$ is a mixture of corpus-wide topics distributions $\overline{\beta}_k$. It assumes the following generative process for each document:

1. Choose a prior distribution $\overline{\theta} \sim \text{Dir}(\overline{\alpha})$ over topics.

2. For each word w_n:

   (a) Choose a topic $z_n \sim \text{Multinomial}(\overline{\theta})$

   (b) Choose $w_n \sim \text{Multinomial}(\overline{\beta}_{z_n})$

One limitation of LDA is that it cannot model correlation between topics even though it is often the

case in practice that certain topics naturally co-occur. This limitations arises from the usage of the Dirichlet distribution to model the variability among the topic proportions. Blei et al. (2007) expand more on the topic, suggesting CTMs, where the topic proportions exhibit correlation via the logistic normal distribution (Atchison and Shen, 1980). It assumes a similar generative process as LDA, except the topic proportions are drawn from a logistic normal, rather than a Dirichlet.

In our case, we represent each document (i.e. textual description of a tender or supplier) as a L-dimensional topic proportion generated from a logistic normal distribution. We compute the distance between suppliers and tenders using the Jensen-Shannon divergence (Lin, 1991) between the corresponding predicted topic proportions. Given a new supplier or new contract, the algorithm recommends top N tenders or suppliers which have lowest distances to the new entity.

We use the term CTM to refer to this approach. When the dimension L of the topic proportion is relevant, we use the naming convention CTM-$\langle$L$\rangle$.

4 Experiments

4.1 Performance Metrics

In this section, we first introduce the metrics that we used to assess the performance of the methods proposed in Section 3 for addressing the SEARCH and RECOMMEND problems. Each paragraph below discusses one problem.

SEARCH Recall from Section 3 that a method which approaches the SEARCH problem, given a supplier s and an integer n, produces a list $\sigma^n(s) = t_{\pi_1} t_{\pi_2} \dots t_{\pi_n}$ of n tenders ordered by their suitability for the services that the supplier offers, where the suitability is indicated by an ordering function π. In our data set, we have knowledge of which tenders were awarded to which suppliers. Let T_s be the set of tenders awarded to supplier s. We define the retrieval-threshold $k \in [0, 1]$ such that $n = \lfloor |T| k \rfloor$, where $\lfloor . \rfloor$ is the *floor* function that takes a real number and returns the largest integer less than or equal to the given real number. We define the *hit-count*

$$R_k^{\text{hit}} = |\{t \in T_s : t \in \sigma^n(s)\}|.$$

That is, the cardinality of the set of true awarded tenders that appear in the sequence $\sigma^n(s)$. Similarly, we define the *miss-count*

$$R_k^{\text{miss}} = |\{t \in T_s : t \in \sigma^{|T|}(s)_{|T|-n:|T|}\}|,$$

where the subscript $|T| - n : |T|$ refers to taking the subsequence from $\sigma^{|T|}(s)$ from position $|T| - n$ to position $|T|$, inclusive. In such a setting, we define the retrieval score metric as follows:

$$R_k = \frac{1}{1 + e^{R_k^{\text{miss}} - R_k^{hit}}}.$$

Figure 5 shows an example of how the retrieval-score is computed.

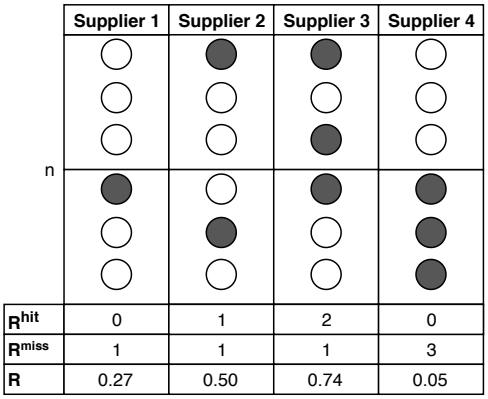

Figure 5: Example computation of retrieval score for 4 suppliers and 6 tenders, as discussed in Section 4.1. Each circle represents a returned tender. Full circles represent correct returns (hits), while empty ones represent incorrect returns (misses).

RECOMMEND Recall from Section 3 that a method which approaches the RECOMMEND problem, given a tender t and an integer n, produces a list $\tau^n(t) = s_{\pi_1} s_{\pi_2} \dots s_{\pi_n}$ of n suppliers ordered by their suitability to be awarded tender t, where the suitability is indicated by an ordering function π. Let $\mathbb{1}\{\mathcal{P}\}$ be the indicator function of the logical proposition $\mathcal{P}$, that is,

$$\mathbb{1}\{\mathcal{P}\} = \begin{cases} 1, & \text{if } \mathcal{P} \text{ is true} \\ 0, & \text{if } \mathcal{P} \text{ is false.} \end{cases}$$

In our data set, we have knowledge of which tenders were awarded to which suppliers. Let s_t be the supplier that undertook tender t. That is, s_t is the supplier that we know has been awarded tender t. We define the hit threshold $k \in [0, 1]$ such that $n = \lfloor |S| k \rfloor$. In such a setting, we define the k-hit-rate metric as follows:

$$H_k(\tau, S, T) = \frac{\sum_{t=1}^{|T|} \mathbb{1}\{s_t \in \tau^n(t)\}}{|T|}.$$

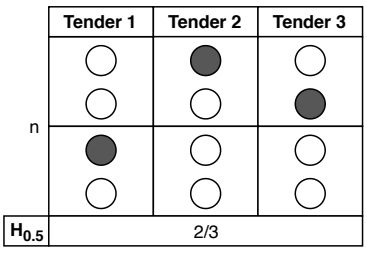

Figure 6: Example computation of k-hit-rate for 4 suppliers and 3 tenders, as discussed in Section 4.1, with $k = 0.5$. Each circle represents a supplier. Full circles represent the target supplier for the respective tender.

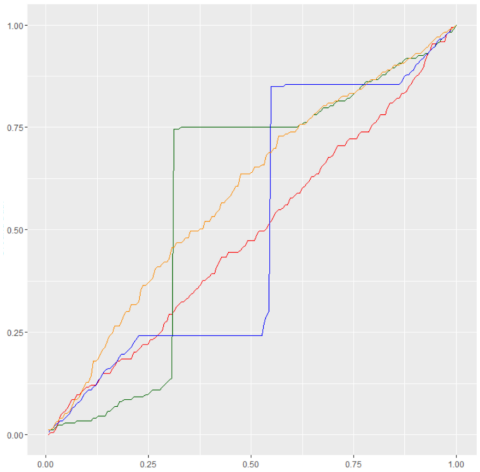

Figure 7: Hit-rate for the RECOMMEND problem achieved by TEXT (red), F-TEXT (green), SDNE (blue) and MERGED (orange), as discussed in Section 4.2. The horizontal and vertical axes corresponds to the hit threshold and the hit-rate, respectively.

Intuitively, the hit-threshold k specifies the ratio of suppliers, out of the total number of existing suppliers, that the recommendation system should return for a given tender t. Then, the k-hit-rate is the ratio of tenders, out of the total number of tenders, for which the target supplier was among the suppliers returned at the hit-threshold k. Figure 6 shows an example of how the k-hit-rate is computed.

4.2 Results

Figure 7 shows a plot of the hit-rate achieved by TEXT, F-TEXT, SDNE and MERGED approaches for hit threshold $k \in [0, 1]$. We notice F-TEXT performs better than TEXT, with a steep jump in the graph which represents recommending the dominating supplier in our dataset, that is, the supplier that was linked to most tenders. SDNE follows a similar trend, but seems to recommend this supplier earlier than F-TEXT.

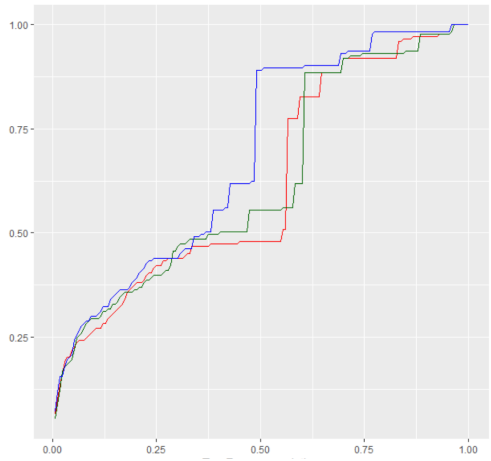

Figure 8: Hit-rate for the RECOMMEND problem achieved by CTM-50 (red), CTM-100 (green) and CTM-200 (blue), as discussed in Section 4.2. The horizontal and vertical axes corresponds to the hit threshold and the hit-rate, respectively.

Finally, MERGED shows not only an improvement over all when k is low, but also seems robust to the dominating supplier effect, as its graph presents no steep jumps. For CTM, Figure 8 shows the hit-rate achieved by CTM-50, CTM-100 and CTM-200 for hit threshold $k \in [0, 1]$. We notice CTM-200 performs the best out of these. Finally, Figure 9 shows the hit-rate for F-TEXT and SDNE, the best performing individual embedding approaches, compared to CTM-200, the best performing topic modelling approach. We find that CTM-200 achieves the best overall performance.

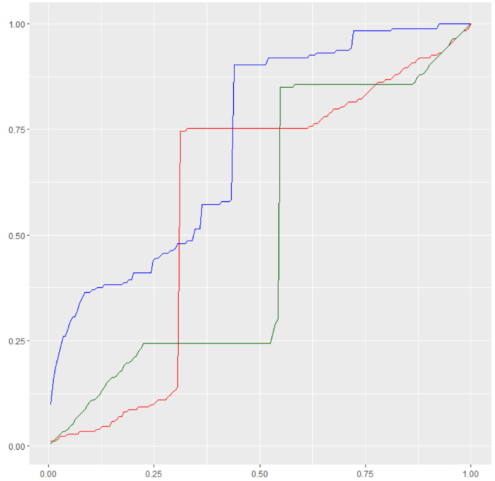

Figure 9: Hit-rate for the RECOMMEND problem achieved by F-TEXT (red), SDNE (green) and CTM-200 (blue), as discussed in Section 4.2. The horizontal and vertical axes corresponds to the hit threshold, and the hit-rate, respectively.

5 Team Members

- Luis Ospina-Forero, a Presidential Fellow in Data Science at the Alliance Manchester Business school. His research interests relate to Network Inference, Network Comparison and the application of Network Analysis methodologies to practical and meaningful problems in finance and society. Luis Ospina-Forero directed the work of the group, together with Mihai Cururingu.

- Mihai Cucuringu, an Associate Professor in the Department of Statistics, and an Affiliate Faculty in the Mathematical Institute at University of Oxford. He is also a Non-Tutorial Fellow at Merton College, University of Oxford; a Turing Fellow at The Alan Turing Institute in London; and a Senior Research Fellow at the Institute for New Economic Thinking at the Oxford Martin School. He is interested in the development and mathematical & statistical analysis of algorithms for data science, network analysis, and certain computationally-hard inverse problems on large graphs, with applications to various problems in machine learning, statistics, and finance, often with an eye towards extracting structure from time-dependent data which can be subsequently leveraged for prediction purposes. He coordinated the work in the group, together with Luis Ospina-Forero.

- Oduwa Edo-Osagie, a PhD student at University of East Anglia focused on machine learning methods for the detection and monitoring of communicable and non-communicable diseases via social media. He contributed to this report by working on the generation of text embeddings helping derive subsequent adjacency matrices from text embeddings.

- Hoang Le, a PhD student at the Trinity College Dublin. His research focuses on Bayesian methods for pair matching problems. He contributed to the report by building the topic modelling model and its recommender system.

- Silviu Oprea, a PhD student in Data Science at the University of Edinburgh. He is interested in computational linguistics and natural language processing, especially related to the analysis of figurative language, such as sarcasm and metaphor. He contributed to building the textual and graph embeddings. He also worked on writing the final version of this report.

- Bemsibom Toh, a PhD student at Heriot-Watt University, Edinburgh. His research focuses on applying the theory of large deviations to the study of queuing systems. He contributed to the report by writing the scripts to compute the similarity and adjacency matrices from the text embeddings and defining performance metrics for the recommendation systems.

- Erin Clark, the representative from Spend Network. She has a background in physics and has joined the company in 2015. Her work was to collect, map and analyse the datasets that were used, and will continue work on the project after the challenge. She oversaw the project by explaining the content and the relevance of the data, and explaining the company goals.

- Aude Vuilliomenet, an MSc student at The Center for Advanced Spatial Analysis at University College London. She is interested in understanding the role of the urban environment on peoples behaviour and currently works on spatial analysis and spatial modelling. She was the facilitator of the group, mainly coordinating the project with the main project researchers (Luis Ospina and Mihai Cucuringu). She contributed to this report by working on the validation for the topic modelling.

- Kai Xu, a PhD student at the University of Edinburgh. His research focuses on probabilistic machine learning, especially approximate methods in Bayesian inference. He contributed the report by exploratory data analysis based on spectral algorithms for graphs with the support from Mihai.

- Yuchen Zhao, a postdoctoral research fellow in the School of Electronics and Computer Science at the University of Southampton. He obtained his PhD in Computer Science from the University of St Andrews. He contributed to the project by data preprocessing, evaluation, and report writing.

References

J Atchison and Sheng M Shen. 1980. Logistic-normal distributions: Some properties and uses. *Biometrika*, 67(2):261–272.

M. Belkin and P. Niyogi. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396.

Steven Bird. 2006. NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, Sydney, Australia. Association for Computational Linguistics.

David M Blei, John D Lafferty, et al. 2007. A correlated topic model of science. *The Annals of Applied Statistics*, 1(1):17–35.

Niladri Sekhar Dash. 2008. Context and contextual word meaning. *SKASE Journal of Theoretical Linguistics*.

Emily M. Jin, Michelle Girvan, and M. E. J. Newman. 2001. Structure of growing social networks. *Phys. Rev. E*, 64:046132.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053.

Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.

D. E. Rumelhart and J. L. McClelland. 1987. *Learning Internal Representations by Error Propagation*, pages 318–362. MITP.

Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1225–1234, New York, NY, USA. ACM.

turing.ac.uk
@turinginst