Mapintel Project Report

David Silva, Fernando Bação

July 3, 2021

Abstract

Briefly summarize your previous work, goals and objectives, what you have accomplished, and future work. (100 words max) If you have a question, please use the help menu ("?") on the top bar to search for help or ask us a question.

Introduction

Competitive Intelligence (CI) is a system of environmental scanning that involves the collection and analysis of information with the objective to achieve competitive advantage. According to Brod (1999), "Companies with competitive intelligence programs have better knowledge of their markets, better cross-functional relationships between their business units and a greater ability to develop proactive competitive strategies." CI has a fundamental role in helping businesses remain competitive, influencing a wide range of decision-making areas, and leading to substantial improvements such as the increase of revenue, new products or services, cost savings, time savings, profit increases, and achievement of financial goals (Calof et al., 2017).

The success of CI comes from two main characteristics: the availability of environmental data and the process of extracting information from such data. The former has seen a significant improvement because of the "digitalization" of the market and business activities. Data about companies' actions and interaction is public and can be leveraged to gain any kind of competitive advantage. However, the latter still remains limited by the capacity of analysts to sift through large volumes of text. In order to scale to the ever-growing dimension of data, the task of mining information about the environment needs to be redesigned without disregarding the important role that analysts play on filtering relevant information and identifying possible business opportunities and risks. Therefore, the goal is to enhance the analyst's task by providing a tool to explore, organize and visualize the environmental data present in the array of existing sources.

A survey made to CI professionals in Marin and Poulter (2004) revealed that the most common sources of CI are, in order of importance, news providers, corporate websites and trade publications and that such information can be obtained from a wide variety of channels such as employees, clients and suppliers. Dey et al. (2011) also shows that social networks contain relevant information, particularly on promotional events and consumer perception towards products, services and brands. CI resources on the web come from a variety of sources, the underlying data is unstructured, and is often accompanied by a considerable amount of noise. These characteristics add to the difficulty of the analyst's task and exacerbate the need for tools to support it.

Various studies have attempted to create systems for exploring and gathering intelligence from large collections of textual data (Ji et al., 2019; Lafia et al., 2019, 2021; Dey et al., 2011). These studies have consistently applied Natural Language Processing (NLP) techniques for helping users comprehend large volumes of text without requiring to sift through every document. Dey et al. (2011) focuses on the designing of a system for CI that captures data from multiple sources, cleans it, uses NLP to identify and tag the relevant content, stores it, generates consolidated reports and can also produce alerts on pre-defined triggers.

Although, the previously mentioned systems have successfully been used for dealing with large amounts of text, insufficient attention has been paid to the CI analyst's task, particularly on the exploratory and investigative aspect of it. Accordingly, we intend to improve the existing systems in two ways: by adding a module of information retrieval that allows to perform ad hoc queries on the document collection, giving the user the ability to accurately satisfy any information need that might emerge, and by building a visual interface that organizes and displays the entire collection, giving the user the ability to explore the data and to focus on particular subsets of documents with thematic commonalities.

In this paper, we explore how state-of-the-art NLP techniques can be used in a system for supporting CI analysts in the process of extracting information from environmental data.

Related Work

We review methods that facilitate the environment scanning task by abstracting and visually summarizing large collections of documents. To situate our contribution, we first complete the review of systems for exploring and gathering intelligence from a text corpus. We then describe the document embedding, dimensionality reduction, and data visualization techniques used to design these systems.

Ji et al. (2019) proposes a system for visual exploration of neural document embeddings to gain insights into the underlying embedding space and to promote the utilization in prevalent IR applications. t-SNE is used to project the high-dimensional data onto a 2D surface. This technique is able to capture both local and global structure from the high-dimensional data in an efficient and reliable way. In this work, the documents are embedded using the Paragraph Vector model. The system visualizes neural document embeddings as a configurable document map and enables guidance and reasoning, facilitates to explore the neural embedding space, identifies salient neural dimensions (semantic features) per task and domain interest and supports advisable feature selection (semantic analysis) along with instant visual feedback to promote IR performance. Overall, the system provides users with insights and confidence in neural document embeddings given their black-box nature.

Lafia et al. (2019) uses SOM and Latent Dirichlet Allocation (LDA) to convey the relatedness of research themes in a multidisciplinary university library. Documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. That said, each document is embedded in a vector space of N dimensions, corresponding to the number of topics selected. SOM produces a landscape for exploring the topic space and provides users with an overview of the document collection and the ability to navigate (discover items of interest), change the level of detail, select individual documents and discover relationships between documents.

Kaski et al. (1998) presents the WEBSOM - a system that organizes a textual document collection using a SOM-based graphical map display that provides an overview of the collection and facilitates interactive browsing. Kohonen (2013) revisits the topic and provides some enhancements. Here, the documents are represented with a TF-IDF weighting and a random projection is used to reduce the dimensionality of the vector space, while preserving the similarity structure between documents. A SOM is constructed and each document is mapped into the node that best represents it. This provides exploring, searching and filtering capabilities. For example, when a node in the map is clicked, the titles of the corresponding documents and eventually some additional information such as descriptive words are presented. Also, the map is described by an automatic annotation procedure explained in Lagus and Kaski (1999), which helps to understand the semantics encoded in each map region. The user can also perform queries either using a set of keywords or a descriptive sentence. The query is then mapped into the reduced vector space and matched with the most similar documents and/or nodes. A zooming feature is also present which allows the user to explore specific regions of the map with finer detail.

Henriques et al. (2012) proposes the GeoSOM suite, a tool for geographic knowledge discovery using SOM. This tool is designed to integrate geographic information and aspatial variables in order to assist the geographic analyst's objectives and needs. The tool provides several dynamically linked views of the data consisting of a geographic map, an u-matrix, component plate plots, hit-map plots, parallel coordinate plots, boxplots and histograms. These views and their connection allows for an interactive exploration of the data.

(Lafia et al., 2021) proposes a method for modeling and mapping topics from bibliometric data and build a web application based on this method. They also perform a user evaluation of the topic map. The map produced allows users to read a body of research "at a distance", while providing multiple levels of detail of the topics that represent the documents. They also incorporate a time dimension, allowing users to understand the evolution of the topics over time. They compare both non-negative matrix factorization (NMF) (Lee and Seung, 1999) and LDA for discovering the underlying topics in the data and obtaining vector representations of the documents. For visualizing these documents, they compare both t-SNE and UMAP. The best performing configuration uses NMF with t-SNE. To allow for different detail levels, the authors produce two maps: a coarse map of 9 topics that gives a general overview of the topics within the data and a detailed map of 36 topics that captures more specific research themes. The web application consists of an interactive dashboard that allows users to explore the map of documents.

Method

The methodology adopted in this project can be summarized by Figure 1. First, we set up a data collection process to automatically absorb the continuous flow of news articles, then some pre-processing was applied to the data to make it usable by the embedding models and improve the quality of the produced document vectors. A SOM was applied to build a two-dimensional grid that is able to represent the multi-dimensional input space and therefore, the properties and relationships of the articles. Finally, a document exploration interface was built to provide the user the ability to explore and query the articles. The code developed for the project can be accessed at github.com/DavidSilva98/mapintel_project.

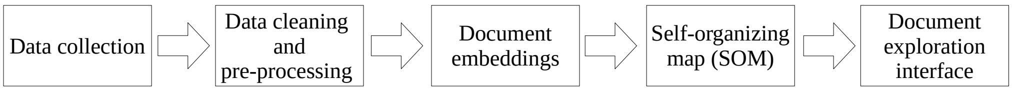

Figure 1: Methodology

Data Collection

In this project we decided to focus on how NLP and particularly sentence embeddings could help in organizing, exploring and retrieving text documents. Since the objective is to explore news articles, we used a REST API [1] to continuously retrieve English articles from multiple international sources several times a day. The API calls are performed through the AWS Lambda service [2] and the articles, as well as their metadata, are stored using the MongoDB Atlas cloud database service [3]. One particularly useful feature of the metadata is the category of the article. This can be one of the following: business, entertainment, general, health, science, sports or technology. It is important to note that the API we are using imposes some limitations that affect the data collection such as the articles being provided with 1 hour delay, having a maximum of 100 requests per day and the content of the article being truncated to 200 characters. We also developed a simple Optical Character Recognition (OCR) pipeline using the Tesseract OCR engine [4]. The purpose of this pipeline was to integrate internal documents from AICEP in our application, however we haven't focused on these documents so far.

Data Cleaning and Pre-processing

After loading the document corpus from the database, we concatenated the title, description and content fields in order to obtain longer and more informative documents. We proceed to clean the documents by removing non-textual patterns such as URLs and HTML tags and by removing non-English articles which are still present despite the filter applied previously. We split the document corpus into train and test set to allow for downstream unbiased performance assessments. A pre-process pipeline is applied on both corpora [5] to reduce the dimensionality of the vocabulary. The pipeline consists of removing stop words (very frequent words that are irrelevant), lowering the letter's case, removing accents and punctuation, applying stemming [6] and removing words that appear in just one document or in more than 90% of the documents.

Document Embeddings

Once the corpus is pre-processed, we encoded each document as a single vector of information. Here we used several approaches and compared them using a standardized evaluation design. As a baseline we used a Bag-of-Words (BOW) approach (Harris, 1954), which represents each document as a token histogram of the top 10000 most frequent tokens. We also used a Term Frequency - Inverse Document-Frequency (TF-IDF) approach (Jones, 1972) which tries to adjust for the fact that some words appear more frequently in general by offsetting each token frequency with the number of documents that contain the token. In these 10000 tokens, we included n-grams containing one to three words so word order information could be included in the embedding. We also used the proposed models by Le and Mikolov

[1] newsapi.org
[2] A serverless compute service that lets you run code without provisioning or managing servers
[3] A fully-managed cloud database service
[4] github.com/tesseract-ocr/tesseract
[5] Note: the pipeline is fitted only on the train set to avoid data leakage
[6] Term normalization process that removes the morphological and inflectional endings from words

(2014), namely the Distributed Memory Model of Paragraph Vectors (PV-DM) and the Distributed Bag of Words version of Paragraph Vector (PV-DBOW), to learn continuous distributed fixed-length vector representations from variable-length pieces of text. These models are trained on the task of predicting words in a paragraph by looking at the context of the target word, which is encoded in the words and paragraph vectors. These vectors are the parameters of the model and are adjusted using stochastic gradient descent and backpropagation. One of the disadvantages of these methods is that to infer the embeddings of new documents, the model needs to train them which can be a problem when dealing with user queries.

Model Evaluation

To evaluate the quality of each approach, we used the corresponding embeddings and categories of each document in various tasks, similarly to some of the ones in Conneau and Kiela (2018). One of the approaches was training a logistic regression model using the embedding vectors of the train corpus to predict the article category. The accuracy of the classifier on the test corpus was used to evaluate the embeddings as the model is kept constant over the different approaches. We realized that, even though the model is kept constant, we cannot control for the interactions between each feature set and the logistic regression, which means the scores obtained don't completely isolate the embeddings performance. For this reason, a second task was proposed which consisted in classifying whether each unique test document pair belonged to the same category, based solely on their cosine similarity. The cosine similarities were converted to a range between 0 and 1 using the min-max transformation and the average binary cross-entropy over all unique pairs of documents was obtained. We also looked at the t-SNE projections of the embeddings to visualize how well they captured the semantics of the documents. Our hope was that if some semantic properties were captured, the embedding vectors would have been grouped by their categories. In the results section we will analyze how the different embedding models produced different evaluations.

Self-Organizing Map

After comparing the several embedding models, the SOM model was applied on the train embeddings of the best performing model using a fork of the SOMPY package (Moosavi et al., 2014). The grid is composed of regular hexagons as they are "visually much more illustrative and accurate, and are recommended" (Kohonen, 2013). Also, we selected the lengths of the horizontal and vertical dimensions of the grid to comply with the relation of the two largest principal components, while providing enough nodes to adequately represent the details and clusters of the input space. The oblong regular arrays have the advantage over the square ones of guaranteeing faster and safer convergence in learning. The nodes were initialized as a regular, two-dimensional sequence of vectors taken along a hyperplane spanned by the two largest principal components of the input space, providing faster ordering and convergence (Kohonen, 2001). Finally, we relied on the minimization of the quantization error (the mean distance of every data point to the corresponding best-matching unit) to select the remaining hyper-parameters of the model.

Document Exploration Interface

We extracted the fitted codebook matrix and we utilized it to build a U-matrix (Ultsch, 1993) to visualize the structures of the high-dimensional input space. By adding an interactive component to this visualization, we were able to encode several details and information within each unit of the matrix such as the distance from its neighbor units, the number of observations allocated to it and also some aggregated information from these observations. We also used the approach in Lagus and Kaski (1999) to characterize regions of the U-matrix by optimal positioning of descriptive keywords. These words function as landmarks i.e. navigational cues that help in maintaining a sense of location during the exploration of the map. Finally, we integrated the remaining components of the interface such as the search bar, a pane to preview the articles retrieved by the query and some more dynamic components to facilitate the user interaction.

Results

Present an observation (results), then explain what happened (analysis). Each paragraph should focus on one aspect of your results. In that same paragraph, you should interpret that result. In other words, there should not be two distinct paragraphs, but instead one paragraph containing one result and the interpretation and analysis of this result. Here are some guiding questions for results and analysis:

When describing your results, present your data, using the guidelines below:

- What happened? What did you find?
- Show your experimental data in a professional way. Refer to Grammar Guidelines for Reports for details on formatting. Be sure to reference figures before they appear in your paper (see Figure 2). Be sure to do the same for tables (see Table 1). For a good tool for making tables, go to tablesgenerator.com.

Figure 2: Captions go beneath figures.

Table 1: Captions go above tables.

Parameter	Symbol	Value
Residence Time	θ	90 s
Hydraulic Gradient	G	500 s^{-1}

After describing a particular result, within a paragraph, go on to connect your work to fundamental physics/chemistry/statics/fluid mechanics, or whatever field is appropriate. Analyze your results and compare with theoretical expectations; or, if you have not yet done the experiments, describe your expectations based on established knowledge. Include implications of your results. How will your results influence the design of AguaClara plants? If possible provide clear recommendations for design changes that should be adopted. Show your experimental data in a professional way using the following guidelines:

- Why did you get those results/data?
- Did these results line up with expectations?
- What went wrong?
- If the data do not support your hypothesis, is there another hypothesis that describes your new data?

Discussion

Study comparison with other studies. What were the limitations?

Conclusions

Explain what you have learned and how that influences your next steps. Why does what you discovered matter to AguaClara? Make sure that you defend your conclusions. (this is conclusions, not opinions!)

Future Work

For implementing the query feature of the system, the query is embedded in the same space of the news article corpus and the distance with each SOM unit is computed. The query is then matched with the closest SOM unit and the documents allocated to that unit are retrieved. This approach is fast since there are many fewer units than documents. The unit's documents are ranked by computing the distance between them and the query. The search quality is expected to not decrease significantly as long as the Mean Quantization Error (MQE) (i.e. the mean euclidean distance each input vector to its BMU) remains low.

We plan to provide a zooming capability on the SOM U-matrix so the user can explore specific regions of the map in detail. There are two ways we have been discussing on how to implement this: one possibility would be to allow the user to select a specific unit or group of units on the map and then provide a projection of the underlying documents using either t-SNE (Van der Maaten and Hinton, 2008) or UMAP (McInnes et al., 2020); a second possibility would be to allow the user to digitally zoom in on the U-matrix, just like it is done in Kaski et al. (1998). An appealing attribute of this option is the preservation of the landmark labels, which are updated according to the zooming of the map.

There's also some discussion on how to integrate release date information on the article's representation. This would allow the documents to be organized not only according to their semantics but also according to their release date. This could also improve the query results as the users are most likely interested on current information. Another feature related to release date would be to relate documents in a time line, allowing a specific subject to be tracked through time.

We would also like to improve the data collection pipeline since we are relying directly on NewsAPI free subscription which has some limitations already described. This would require a substantial effort since web scrapping would most likely be the necessary solution. This approach would provide us with the full article content and would allow us to collect articles as soon as they are released. Multilingual articles could also be collected and integrated into the system by using multilingual embeddings models such as Conneau et al. (2019).

Some more ideas to explore consist on: build a single or multi article summary feature, to provide a brief resume of the content of a specific article or of a specific SOM unit (collection of articles); add a news article feed based on individual user viewing history. If we plan to expand the application to multiple users, an implicit feedback collaborative filtering (Hu et al., 2008) approach could be used.

Some research on understanding the document embedding dimensions' would also be interesting as these usually present a correlation structure which captures the latent semantical topics of the document collection as seen in Ji et al. (2019). This would provide the user with the necessary confidence on the neural document embeddings that is lacking because of the black-box nature of these models.

References

Brod, S. (1999). Competitive intelligence: Harvesting information to compete and market intelligently. *Camares Communications, New York, NY.*

Calof, J., Sewdass, N., and Arcos, R. (2017). Competitive Intelligence: A 10-year Global Development. Technical report, Competitive Intelligence Foundation.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116.*

Conneau, A. and Kiela, D. (2018). Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449.*

Dey, L., Haque, S. M., Khurdiya, A., and Shroff, G. (2011). Acquiring competitive intelligence from social media. In *Proceedings of the 2011 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data*, MOCR_AND '11, pages 1–9, New York, NY, USA. Association for Computing Machinery.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

Henriques, R., Bacao, F., and Lobo, V. (2012). Exploratory geospatial data analysis using the GeoSOM suite. *Computers, Environment and Urban Systems*, 36(3):218–232.

Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272. Ieee.

Ji, X., Shen, H.-W., Ritter, A., Machiraju, R., and Yen, P.-Y. (2019). Visual exploration of neural document embedding in information retrieval: Semantics and feature selection. *IEEE transactions on visualization and computer graphics*, 25(6):2181–2192.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Kaski, S., Honkela, T., Lagus, K., and Kohonen, T. (1998). WEBSOM–self-organizing maps of document collections. *Neurocomputing*, 21(1-3):101–117.

Kohonen, T. (2001). Software Tools for SOM. In *Self-Organizing Maps*, pages 311–328. Springer.

Kohonen, T. (2013). Essentials of the self-organizing map. *Neural networks*, 37:52–65.

Lafia, S., Kuhn, W., Caylor, K., and Hemphill, L. (2021). Mapping research topics at multiple levels of detail. *Patterns*, 2(3):100210.

Lafia, S., Last, C., and Kuhn, W. (2019). Enabling the Discovery of Thematically Related Research Objects with Systematic Spatializations. In *14th International Conference on Spatial Information Theory (COSIT 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Lagus, K. and Kaski, S. (1999). Keyword selection method for characterizing text document maps.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196. PMLR.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

Marin, J. and Poulter, A. (2004). Dissemination of Competitive Intelligence. *Journal of Information Science*, 30(2):165–180.

McInnes, L., Healy, J., and Melville, J. (2020). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv:1802.03426 [cs, stat]*.

Moosavi, V., Packmann, S., and Vallés, I. (2014). SOMPY: A Python Library for Self Organizing Map (SOM).

Ultsch, A. (1993). Self-organizing neural networks for visualisation and classification. In *Information and Classification*, pages 307–313. Springer.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).