Enhancing Competitive Intelligence Acquisition Through Embeddings

David Silva, Fernando Bação

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).

We argue that 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 interactions are public and can be leveraged to gain any kind of competitive advantage. However, the latter 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 in 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 aspects of it. Accordingly, we intend to improve the existing systems in two ways: by adding a module of information retrieval that allows performing 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 find subsets of documents with thematic commonalities and to interactively explore the data while promoting serendipity.

In this paper, we explore how recent advancements in NLP, namely in representing documents as semantic vectors called embeddings, 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 can capture both local and global structures from high-dimensional data efficiently and reliably. 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 allow 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 capture more specific research themes. The web application consists of an interactive dashboard that allows users to explore the map of documents.

Method

We propose a system that supports the exploration of a document collection while sensemaking and can satisfy emerging information needs by allowing full-text queries over the entire collection. The system is scalable to large amounts of data, is dynamic as it regularly integrates new data, and is fast. It is composed of three main pipelines: Indexing, Query, and Visualization which objectives are respectively, to get documents and their metadata from a source to a database, to retrieve the most relevant results to a user query, and to produce an interactive interface for exploring the document collection. The code developed for this work can be accessed at github.com/DavidSilva98/mapintel_project.

Indexing

In this work we decided to focus on how NLP and particularly sentence embeddings could help in organizing, exploring, and retrieving text documents in the CI domain. As already stated, there are multiple sources of CI, and different information can be obtained from these. Dey et al. (2011) shows in Table 1 what kind of information can be acquired from these sources, particularly the ones that are easily available through the web. We decided to work mainly with news articles as they provide a general and accessible means of information about the environment, however, our methodology is easily extensible to data from different sources and can be applied in various settings.

Table 1: Competitive Intelligence resources on the web

Type of Competitive Intelligence	**Web resources**
People events	News, company web-sites
Competitor strategies, Technology investment, etc.	News, Discussion forum, Blogs, Patent search sites
Consumer sentiments	Review sites, Social networking sites
Promotional events and pricing	Social networking sites
Related real-world events	News, Social networking sites

To obtain news articles from multiple international sources, we use a REST API[1]. The API retrieves the articles, as well as their metadata, consisting of attributes such as source, author, title, description, content, category, URL, and publication date and time. We use this API to feed the system with updated data on a schedule while focusing on articles written in English from a set of predefined categories (business, entertainment, general, health, science, sports, technology).

Due to API limitations, the retrieved data has its content attribute truncated to 200 characters. To overcome this, we treat a single document unit as the concatenation of title, description, and content, providing us a semantically sufficient piece of text that we can use for NLP purposes. Despite this limitation, we give the user the possibility of accessing the full article through its URL. We ensure that each document is unique, is written in English, and doesn't have any HTML tags or any strange pattern.

After, we produce the embeddings of each document. This process is the basis of our work as it allows us to encode the semantic identity of the article onto a vector of a given dimensionality. This semantic identity describes what is the subject of the article, and it can be used to compare documents between each other i.e. articles with the same subject will be close in the semantic space and vice-versa. We use SBERT (Reimers and Gurevych, 2019) to embed the documents using a pre-trained encoder trained on the MS MARCO dataset (Bajaj et al., 2018), a large-scale information retrieval corpus consisting of about 500,000 real search queries from Bing search engine with the relevant text passage that answers the query. This produces vectors of 768 dimensions, which we then reduce to 2 dimensions using the UMAP (McInnes et al., 2020) algorithm. UMAP constructs a topological representation of the high and low dimensional data and its goal is to minimize their cross-entropy, which measures the difference between the two representations, by adjusting the low-dimensional representation. This is another important

[1] newsapi.org

component of our system as it allows the organization and localization of the entire document collection in a 2-dimensional map, which can be used to explore and interact with the data. We opted to use UMAP over other dimensionality reduction techniques because of its improved map quality, reduction in time required to produce the output map, support of larger data set sizes, and, most importantly, its ability to update the output map with new data without having to rebuild it (McInnes et al., 2020).

Finally, we load the documents, their metadata, their SBERT embeddings, and their UMAP embeddings into a database. We use Open Distro for Elasticsearch[2] — an open-source, RESTful, distributed search and analytics engine based on Apache Lucene[3] — to store the data, organize it in an index and perform full-text search on it. We can think of the approach described as an Indexing Pipeline — Figure 1 — that extracts new raw documents from a data source, pre-processes and manipulates them, stores the results in a database, and indexes the documents for future search tasks.

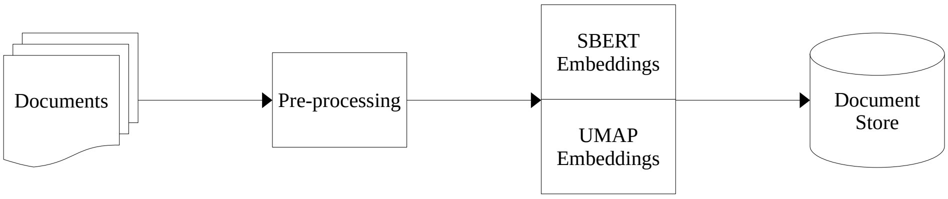

Figure 1: Indexing Pipeline

Query

Finding meaningful information within a large amount of data is a big part of the CI task. The ability to retrieve relevant documents from a large collection of news articles through natural language queries empowers the CI analyst with an easy and intuitive interface to scan the environment.

Our system provides a search functionality based on Open Distro for Elasticsearch and its k-Nearest Neighbor (k-NN) Search module. By utilizing the k-NN module, we can leverage the SBERT embeddings by projecting the query string onto the same semantic space as the corpus and computing its k-nearest neighbors i.e. finding the k documents whose embedding vectors are closest to the query embedding vector, according to some pre-defined similarity metric. Since the embedding vectors encode the semantic identity of each document, this method provides semantically relevant results for a given query. Furthermore, the k-NN module delivers a highly performant and scalable similarity search engine by leveraging Elasticsearch's distributed architecture and by implementing Approximate Nearest Neighbors (ANN) search based on Hierarchical Navigable Small World Graphs (Malkov and Yashunin, 2018). The k-NN module can also be combined with binary filters that help the user obtain focused results based on characteristics of the documents such as publication date and category. These filters are applied directly to the database, reducing the search space as a result and improving the subsequent search time.

Once again, we can think of the search functionality as a pipeline, illustrated in Figure 2, where we feed a query string and some binary filters, and we obtain documents ordered by their relevancy to the query. We employ a Retrieve and Re-rank pipeline based on the work of Nogueira and Cho (2020); Kratzwald et al. (2019) composed by a "Retrieval Bi-Encoder + ANN" node that performs semantic search using Elasticsearch's k-NN module as described above, and by a "Re-Ranker Cross-Encoder" node consisting of a BERT (Devlin et al., 2019) model fine-tuned on the MS MARCO dataset that receives a document and query pair as input and predicts the probability of the document being relevant.

The pipeline works by taking advantage of the characteristics of both nodes. The Bi-Encoder together with ANN search can retrieve fairly relevant candidates while dealing efficiently with a large collection of documents. The Cross-Encoder isn't as efficient since it has to be performed independently for each document, given a query. However, since attention is performed across the query and the document, the performance is higher in the second node. Therefore, we combine both nodes by retrieving a large set of candidates from the entire collection using the Bi-Encoder, and by filtering the most relevant candidates with the Cross-Encoder.

With this pipeline, we can provide relevant documents to the user given a query and binary filters while ranking them according to a relevancy score. The pipeline is efficient and makes use of the SBERT embeddings and the Elasticsearch architecture. As an additional feature, we can input a document instead of a query, allowing us to search for semantically similar documents within the collection.

[2] opendistro.github.io/for-elasticsearch

[3] lucene.apache.org

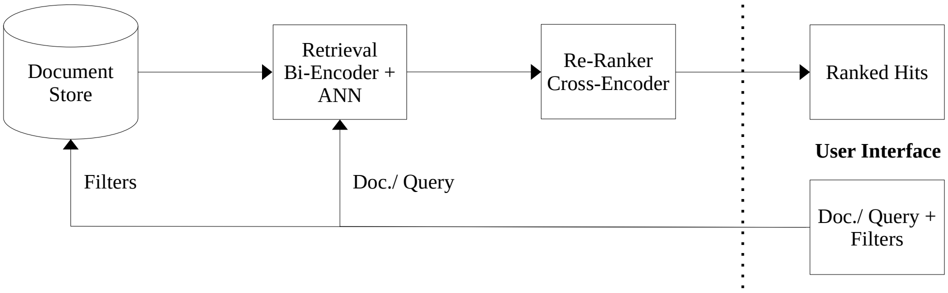

Figure 2: Query Pipeline

Visualization

To facilitate the environment scanning task, we developed a visual interface that organizes and displays the news articles, giving the user the ability to explore the data and zoom on particular regions of the semantic space. The interface uses the UMAP (McInnes et al., 2020) algorithm to reduce the dimensionality of the original semantic space to a 2-dimensional representation that reliably preserves the original topology.

The methodology employed to produce the interface is described in Figure 3. It begins by taking the same inputs passed to the Query pipeline, a query, and a set of filters. The common inputs create a connection between the two modules — when the user queries the database, the query text is projected onto the 2-dimensional map and the filters define which points are displayed in the map. In this way, the map can be seen as a graphical extension of the searching mechanism, where the relevant results reside in the neighborhood of the query, giving the user some insight into how the results are obtained. In addition to the common inputs, we require a relative sample size that defines the percentage of randomly chosen documents (after applying the filters) to be displayed in the map. This is necessary as interaction with the map is hindered by a large number of data points, resulting in a slow and unresponsive experience. Notice that the sample size doesn't affect the query results, as it is always performed on the entire collection.

To produce the interactive scatter plot, the filters and sample size are used to select the documents to be displayed from the database. We compute the SBERT embedding of the query, followed by its UMAP embedding, thus being able to locate the query in the same space as the documents. An advantage of these two models is that we can efficiently produce embeddings of new text without having to re-train them, making this process quite fast. Once we have the UMAP embedding of the query, we join it with the pre-computed embeddings of the documents from the Indexing pipeline, and we produce the interactive map.

The map provides a means to explore the news articles and the different semantic cohorts present within the collection. We color-code the points with the documents' category, and when hovered, the points display their corresponding text. The map can be panned, zoomed in/out and points can be selected.

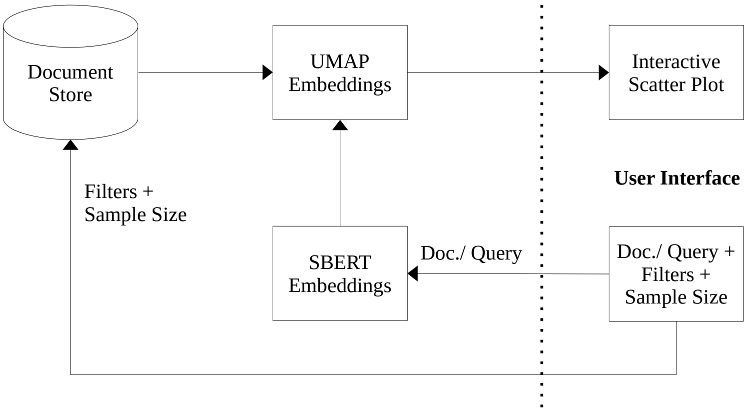

Figure 3: Visualization Pipeline

Results and Discussion

Display the UMAP plot and analyze it.

Apply Top2Vec and analyze the distribution of the topics in the UMAP and the corresponding clusters and compare them with the real categories. Check if the true labels are captured by the clusters i.e. if the clusters contain all the points of a category without containing other categories' points.

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

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 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 in current information. Another feature related to release date would be to relate documents in a timeline, 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 of: build a single or multi-article summary feature, providing a brief resume of the content of a specific article or 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.

References

Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., Rosenberg, M., Song, X., Stoica, A., Tiwary, S., and Wang, T. (2018). MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv:1611.09268 [cs]*.

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*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*.

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.

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.

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

Kratzwald, B., Eigenmann, A., and Feuerriegel, S. (2019). RankQA: Neural Question Answering with Answer Re-Ranking. *arXiv:1906.03008 [cs]*.

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.

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

Malkov, Y. A. and Yashunin, D. A. (2018). Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. *arXiv:1603.09320 [cs]*.

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]*.

Nogueira, R. and Cho, K. (2020). Passage Re-ranking with BERT. *arXiv:1901.04085 [cs]*.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv:1908.10084 [cs]*.

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