Big Data
Volume 00, Number 00, 2023

DOI: 10.1089/big.2022.0215

Automated Natural Language Processing-Based Supplier Discovery for Financial Services

Mauro Papa,[1,2] Ioannis Chatzigiannakis,[2,*] and Aris Anagnostopoulos[2]

Abstract

Public procurement is viewed as a major market force that can be used to promote innovation and drive small and medium-sized enterprises growth. In such cases, procurement system design relies on intermediates that provide vertical linkages between suppliers and providers of innovative services and products. In this work we propose an innovative methodology for decision support in the process of supplier discovery, which precedes the final supplier selection. We focus on data gathered from community-based sources such as Reddit and Wikidata and avoid any use of historical open procurement datasets to identify small and medium sized suppliers of innovative products and services that own very little market shares. We look into a real-world procurement case study from the financial sector focusing on the Financial and Market Data offering and develop an interactive web-based support tool to address certain requirements of the Italian central bank. We demonstrate how a suitable selection of natural language processing models, such as a part-of-speech tagger and a word-embedding model, in combination with a novel named-entity–disambiguation algorithm, can efficiently analyze huge quantity of textual data, increasing the probability of a full coverage of the market.

Keywords: supplier discovery; financial services; entity linking; natural language processing; named-entity–disambiguation

Introduction

Procurement is a strategic process for businesses to acquire products and services, and as such, the criteria for selecting suppliers are important aspects in procurement system design. For enterprises owned by the public, as well as public organizations and the government, public procurement is also viewed as a major market force used as a lever to drive economic growth and achieve objectives such as small and medium-sized enterprises (SMEs) participation and delivering sustainable outcomes, as stated in Organisation for Economic Co-operation and Development (OECD).[1,2] At the same time, according to Brammer and Walker and Pro Inno Europe,[3,4] public procurement is viewed as a direct tool to promote innovation. In such circumstances, the criteria for selecting suppliers are developed in a way that goods or services with unique and innovative elements are favored as pointed out by Georghiou et al.[5]

The criteria for selecting suppliers may range from being well specified in open competitive auctions to being vaguely defined in privately contractual transactions. On one side of the spectrum, open competitive auctions encourage transparent use of public funds, avoid corruption, and help acquire fair pricing, while on the other side of the spectrum, privately-contracted purchases are less formal procedures that lead to improved efficiency in finding specialized services and products as indicated in OECD.[6] Between these two extremes, for example, as in the case of the Italian public sector reported in Edquist and Hommen and Baltrunaite et al.,[7,8] procurement systems foresee intermediates that provide vertical linkages between suppliers and providers.

[1]Bank of Italy, Rome, Italy.
[2]Department of Computer, Control and Management Engineering (DIAG), Sapienza University of Rome, Rome, Italy.

*Address correspondence to: *Ioannis Chatzigiannakis, Department of Computer, Control and Management Engineering (DIAG), Sapienza University of Rome via Ariosto 25, Rome 00185, Italy,* E-mail: ichatz@diag.uniroma1.it

Interestingly, the notion of an intermediary that helps the acquisition of services and commodities dates back to the 16th century in the wool and textile sectors as mentioned in Smith.[9] Since then,[10] note that the role of the intermediary has evolved into an informal disseminator of knowledge about novel technologies.

Intermediaries are specialized service providers undertaking information scanning and technology intelligence, providing foresight and diagnostics role as analyzed in Howells.[10] As a result, innovation intermediaries might be thought of as knowledge brokers. They employ teams of analysts conducting desk research, including interviews with domain experts, product demos, questionnaires, and other methods, to identify new goods and services and grade each vendor against a set of rigorous criteria. The most famous of such market research reports are the magic quadrants by Gartner,[†] which are not intended to be an exhaustive analysis of every vendor in a market, but rather a focused analysis.

More recently, with the growing adoption of crowdsourcing techniques to gather information, some companies have developed peer-review sites. According to Nambisan et al.,[11] such sites may provide hints on product alternatives and competitors based on information collected from a community of users and act as a distributed data collection mechanism to reduce the risk of biased results. Such companies include G2, TrustRadius, and Gartner itself via Gartner Peer Insights.[‡]

In both traditional and more recent peer-to-peer approaches, market research is conducted proactively and the extent to which it covers the entire market is directly proportional to the resources available to the team of analysts and the size of the community of peers. In Obwegeser and Müller,[12] it is reported that in most cases it is very resource-consuming and in this sense extremely difficult to identify small and medium-sized suppliers that own very little market shares and are usually excluded from large public procurement tenders.

The starting point of this work is two observations indicated in the relevant literature in terms of innovation management in the context of public procurements: (1) social media offer opportunities to support the gathering of information, yet how they can be integrated into the process of discovering suppliers of innovative products and services still eludes and more guidance on how to develop new tools is requested from the research community as suggested by Muninger et al.;[13] (2) using recent technological advancements and Artificial Intelligence (AI) techniques can be beneficial in procurement selection as pointed out by Obwegeser and Müller and Lewis et al.[12,14] Given these two observations, a generic methodology is proposed that passively monitors publicly available conversations conducted in focused discussion groups on social media to identify data sources that can provide valuable insights about innovative and emerging products and services.

The proposed methodology incorporates an appropriate combination of AI tools from the field of Natural Language Processing (NLP) to analyze the identified data sources and automatically extract knowledge. The identified data along with the extracted knowledge are combined to form an interactive environment to assist the team of analysis during their desk research. Starting from a specific, already-known product, the user can interact with the extracted knowledge base to explore similar products that were previously unknown. Therefore, the proposed methodology does not explicitly rely on networks of peers and does not use historical open procurement datasets to identify suppliers. In this sense, the focus of this work is on the supplier-discovery part of the procurement process, supporting market research and helping to overcome disadvantages in terms of resources and capabilities that hinder innovative SME participation in public procurements.

The core elements of the generic methodology are presented in detail in Generic Methodology for Supplier Discovery section, providing technical insights on the combination of different NLP models to achieve part-of-speech (POS) tagging, word-embedding, and named-entity–disambiguation (NED). Emphasis is given to the overall ability of the methodology to scale to very large quantities of textual data, allowing it to process content generated on social media and identify new suppliers in a short time.

To provide a better understanding of the methodology, we apply it to a real-world procurement scenario for the Financial sector, focusing on the certain requirements of the Italian central bank (Bank of Italy). The developed application gathers information from community-based sources such as Reddit and Wikidata to identify vendors of products and services related to real-time market data, historical tick series, news feeds, and general market offerings that support decision-

[†]https://www.gartner.com/en/documents/3956304/how-markets-and-vendors-are-evaluated-in-gartner-magic-q
[‡]https://analystrelations.org/2015/07/15/can-it-research-be-crowdsourced

making to carry out financial investments. The resulting market research tool provides insights about market players, without relying on information collected from analysts or peers. All the details on the application of the methodology in the Financial sector and the resulting tool are presented in Case Study in the Financial Sector section.

We conduct a detailed qualitative and quantitative evaluation of the developed tool also involving a group of experts from the Bank of Italy. The evaluation demonstrates the benefits of extracting knowledge passively from data that have already been published autonomously on the internet and highlights the capability of the proposed methodology to identify data sources from which it is possible to acquire information that can potentially be of value in the process of supplier discovery. The discussion on the evaluation can be found in Performance Evaluation section. The article discusses the implications of applying the proposed methodology to tender procedures, highlights the benefits, and presents the limitations of the approach. Remark that the views expressed in this work are those of the authors and do not involve the responsibility of the Bank of Italy.

Related Work

In this section we present a series of tools for supplier discovery that have been recently presented in the relevant literature. The selected tools rely on recently developed techniques for automatic supplier discovery within a procurement process using NLP methods that have been studied recently. The first such tool is presented in Futia et al.,[15] which builds a semantic Knowledge Graph (KG) from open procurement data of multiple Italian administrations. The resulting tool allows easy analysis of all issued public contracts and is queryable by the users as a subscription-based search engine.[§] A method based on NLP techniques is used in Aravena-Díaz et al.[16] to identify prospective bidding candidates from a historical application dataset of procurement documents for Chile's Public Procurement System. The system introduced in Haanpää[17] combines word embedding techniques and clustering algorithms on a private procurement database to produce procurement analytics. More recently, Yaozu and Jiangen[18] present a government procurement information platform that relies on the publicly accessible government procurement website of the Hebei province to construct a KG. The so-called government procurement knowledge graph (GPKG) is utilized to create a search engine for human–computer interaction.

A different approach is used by Soylu et al.,[19] that uses semantic technologies to build a KG, including procurement and company data gathered from multiple public European sources that are integrated through a common ontology. Building on top of this KG, a Vendor Intelligence Procurement Solution (VIPS) is proposed that is called TheyBuyForYou (TBFY),[**] which also includes some supplier discovery functionalities. The TBFY platform has been used in different contexts, and the experiences gained are reported in Soylu et al.[20] For example, Soylu et al.[21] report how the common ontology provided by TBFY was used to produce high quality procurement data in Slovenia. The procurement-related data produced by the TBFY platform are combined with company data on the other hand (e.g., registered organization, address, site) using semantic technologies to construct a KG in Guasch et al.[22] The TITAN system introduced in Benítez-Hidalgo et al.[23] follows a similar approach to the latter one that can be applied to a broader context.

A common factor of all the above works is that they rely on one or more historical procurement databases, either public or private, and most of them combine them with company data acquired from business registries. In contrast to the aforementioned tools, our goal is to identify newly introduced products and services that might not have been part of previous contracts or expenditure data. In this work, instead, we want to extract knowledge from textual data collected from a community-based data source such as Reddit. We therefore believe that our work provides evidence on how such a community-based dataset can be analyzed to extract useful insights for procurement purposes. The different approaches implemented by the different tools are summarized in Table 1.

The tools presented above either retrieve company data from existing business registries or rely on the analysis of textual data for the identification of product and service providers. The problem of Entity matching refers to the identification of variations in textual data that refer to the same real-world entity as presented in Barlaug and Gulla.[24] In their work, they survey a variety of different approaches that have been proposed

[§]https://contrattipubblici.org

[**]https://theybuyforyou.eu/business-cases

Table 1. Comparison of closely related tools with the one presented here in terms of the data used to construct the knowledge graph, use of natural language processing-based methods to analyze the knowledge graph, use of named-entity–disambiguation techniques, and use of a query processor

Tool	*KG*	*NLP*	*NED*	*Query processor*
(15)	Open procurement data of multiple Italian administrations	No	No	No
(16)	Historical application dataset of procurement documents for Chile's Public Procurement System	Latent Semantic Analysis	No	Yes
(17)	Private procurement database	Word embedding and HDBScan	No	No
(19)	Multiple public European sources	No	No	No
(18)	Public government procurement website of Hebei province	No	No	Yes
(23)	Multiple public European sources	No	No	No
(22)	Multiple public procurement-related data and company data	No	No	No
Our tool	Community-based data source such as Reddit	Word embedding and USE	Wikidata	Yes

KG, knowledge graph; NED, named-entity–disambiguation; NLP, natural language processing; USE, Universal Sentence Encoder.

during the past years and report that it still remains a challenging problem. The use of *NED* techniques has been investigated in the work of Parravicini et al.,[25] which proposes a framework for entity linking that leverages graph embeddings to perform collective disambiguation. In their work, a graph of entities is precomputed based on data collected from DBpedia that is loaded in memory for real-time disambiguation.

A different approach is proposed by Lin et al.[26] that combines a bidirectional long short-term memory network (Bi-LSTM) with a neural-encoded mention hypergraph, resulting in a highly-scalable model that is capable of recognizing nested-structure mention entities. In Lin et al.[27] the slightly different task of sequence labeling is examined, as a preprocessing step for NED. They propose two frameworks of latent variable conditional random field models, which use the encoding schema as a latent variable to capture the latent structure of the hidden variables and the observed data. The performance of these two models largely depends on hand-craft features which result in poor robustness over different sequence labeling tasks and datasets. To overcome these shortcomings, Shao et al.[28] propose self-attention based models to automatically extract features and achieve higher performance. A different approach is followed in Lin et al.[29] where an attention segmental recurrent neural network that relies on a hierarchical attention neural semi-Markov conditional random field model is proposed for the task of sequence labeling. The hierarchical structure allows to differentiate more important information from less important information when constructing the segmental representation.

Unlike the identification of authors in textual data, or in general identification of individual names, organizational entity names and product names are not randomly assigned but instead are produced as a result of a marketing process, as investigated in Engel et al. and Stoyneva and Vracheva.[30,31] Regarding young enterprises and start-ups,[32] it is observed that business activated in similar areas will use similar terms when deciding their service and product names. This results in the frequent use of particular phrases in numerous companies, services, and product names. The term "onomastic profusion" is coined by Belz et al.[33] to characterize this phenomenon. This phenomenon creates certain limitations to the aforementioned deep neural network based methods. Moreover, sufficiently large corpus of service and product names is currently not available to efficiently train such models.

An attempt to overcome this problem is presented in Primpeli et al.[34] where the WDC Training Dataset for Large-Scale Product Matching is presented, including products listed in e-shops. The different nature of this special version of the NED problem is also noted by Basile et al.[35] They attempt to overcome the problem using a dataset by extracting the beneficiary names recorded in SWIFT (Society for Worldwide Interbank Financial Telecommunications) bank transfers.

In this work, we wish to establish a methodology that can be applied also in markets with fast-changing offerings. In such markets, the dataset of company names is fast changing and therefore the constructed KG cannot be considered static. One might suggest to reconstruct the KG periodically after retraining the underlying deep neural network models. We propose to avoid scraping the KG entirely and instead build entity embeddings at runtime to provide always up-to-date information. Although this approach is slower, by avoiding the need to keep a local copy of the KG, we can use a much larger KG. For this reason, we choose Wikidata instead of DBpedia, a much bigger KG in terms of the number of entities included as indicated in Färber and Rettinger.[36]

For the specific use case presented here, the generation of entity embeddings is based on the Google Universal Sentence Encoder (USE) instead of the PageRank algorithm used by Parravicini et al.[25] since Google USE achieves state-of-the-art results in sentence similarity if compared to the latest NLP models as reported in Horan.[37] Moreover, because of the domain-dependant nature of this work, our methodology may use filters over some properties of Wikidata entities to increase the quality of the results provided to the user.

Generic Methodology for Supplier Discovery

The methodology proposed comprises three phases: (1) the first phase starts with the acquisition of data from social media, either using a combination of web crawling and data scraping techniques or accessing the data through an Application Programming Interface (API) provided by the social media operator; (2) in the sequel, the data are preprocessed and analyzed using a NLP model for POS tagging to extract NOUNS to identify products and services; (3) the resulting data are analyzed further using word-embedding techniques to train search-by-similarity models to compute a similarity rating between the identified offerings; (4) in the fourth phase, relying on information available on the internet the identified products and services along with their supplier are validated. The user is now ready to query the extracted knowledge based on the specific procurement requirements and interactively explore the analyzed data.

The four phases of the research methodology introduced are depicted in Figure 1. The components that are used for each of the four phases are described in more detail in the following sections.

Identification of data sources and acquisition

The first phase starts with an initial identification of text-based data sources that contain opinions, descriptions, and comparisons of the products and vendors of interest. Such kind of text-based data sources includes, for example, Reddit[††] and LinkedIn.[‡‡] Usually, social media include topic-based classification mechanisms that allow us to easily narrow down useful data. In the case of Reddit, discussions are organized into user-created areas of interest called subreddits. Similarly, on LinkedIn, users create groups of interest to organize content connected to specific areas of interest.

After the data sources are identified, web crawling and data scraping techniques can be used to extract the text properly and remove any encoding or serialization from the text. Alternatively, data can be collected using an API. For example, in the case of Reddit, in Baumgartner et al.[38] the Pushshift API is introduced that can be used to retrieve a copy of all Reddit comments and submissions since 2015.

Remark that after the first iteration of the three phases of the methodology, this phase may continue by introducing additional data sources and/or by collecting the latest posts and comments. In this sense, the primary source of raw data can continuously grow over time.

Data preprocessing

It is common for products and services to use complex names that are composed of two or more words. Usually, the NLP models that analyze the data acquired to extract information relevant to products and services work by examining each word and searching for relations among words within the entire text corpus. Consider the example of Figure 2a, where the product name "six financial" may end up being analyzed as two independent words, thus failing to identify it properly. It is therefore necessary to introduce a preprocessing step to address this problem.

In Related Work section we provide a brief discussion of how this problem is resolved using some characteristic methods from the relevant literature. For a thorough review of Neural Network-based approaches we refer to the survey of Barlaug and Gulla.[24] In the relevant literature, this problem is generally addressed by performing simple arithmetic operations (e.g., average, sum) on word vectors or developing an NLP model that instead of working on single words operates on multiword tokens. If one follows the second approach for the example of "six financial," Figure 2b depicts how the multiword tokens are formed thus allowing us to treat the words individually but also as a single entity.

The methodology presented here creates multiword tokens based on a POS model that analyzes entire sentences and categorizes each word in correspondence with how they are used, also depending on the word and its context. Using the POS model, all words characterized as NOUNS are examined and those that appear next to each other are merged into a single word using an underscore. In the example of Figure 2, this results in "six_financial." In this way, the NLP models are forced to treat such product names as single words.

[††]https://en.wikipedia.org/wiki/Reddit
[‡‡]https://en.wikipedia.org/wiki/LinkedIn

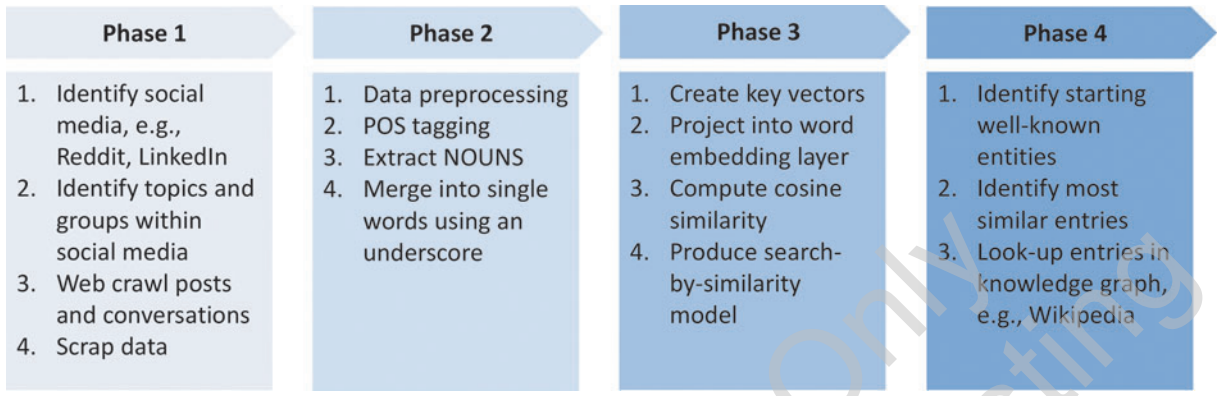

FIG. 1. Basic phases of the research methodology followed.

Text encoding

The next step examines the words identified in the previous phase and their position within the raw text corpus to capture semantic and syntactic features of the corresponding words without human intervention or language-dependent processing. The goal of this phase is to identify the semantic similarity between each word. As indicated in Bengio et al.,[39] text encoding is therefore viewed as an unsupervised learning problem and can be solved using neural network language models.

In more detail, the words produced in the previous phase are converted into key vectors that are projected onto a so-called *word embedding layer* before being fed into other layers of the network. The neural network language model is used to identify a suitable distance metric that can accurately express the semantic similarity between words. Therefore, it is expected that two vectors that have the same meaning will have a similar representation. This is achieved by computing the cosine similarity between each pair of word vectors. The cosine similarity is defined as the cosine of the angle between a pair of vectors. Remark that different similarity measures are available in the relevant literature, here cosine similarity is selected due to the sparse nature of the particular vector space.

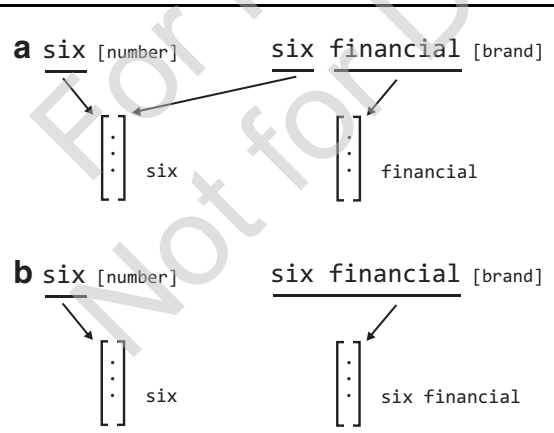

FIG. 2. Analyzing the text without a preprocessing step. **(a)** Without preprocessing: every word is analyzed as independent token. **(b)** Desired outcome: multiword name is converted into a single token.

The resulting model is then used to provide a search-by-similarity feature that allows retrieving the most similar terms for a given word. In this way, starting from previously known products and services, a list of similar words is identified, among which we may discover new products or services previously unknown.

Named-entity recognition and disambiguation

The result obtained in the previous step using the search-by-similarity feature is a list of similar terms that the user has to manually evaluate further to verify the relevance of each term. A central goal of the work presented here is to minimize the human effort by providing additional information that will facilitate the evaluation of a word identified as "similar." Such additional information may potentially include descriptions, official website links, and similarity scores. Therefore in this phase potentially new products and services are identified using a *named-entity recognition and disambiguation* model that maps words of interest against entities in a target KG. The information of the

entities appearing in the KG is used to automatically remove less relevant entities using a suitably selected similarity score.

In the relevant literature, NED algorithms require as input an entire sentence. In this work, instead, the list of semantically similar words produced in the previous phases is used as the input for the NED algorithm. This is done since the goal is to identify the names of products and services within a very specific context. In this particular usage of the NED algorithms, the words need to refer to the same context, for example, computing products and services, and this is an important element that can be used to improve the performance of the methodology.

Consider the example depicted in Figure 3, where the word "Apple" needs to be disambiguated using Wikidata as the KG. After consulting the KG, two candidate entities emerge: "Apple, Inc.," and "Apple (fruit)." If the word-embedding model returned "Apple" as a similar word for "Microsoft," then with high probability we would be referring to "Apple, Inc." If, instead, the word-embedding model returned "Apple" as a similar word for "Orange," then probably we would be referring to "Apple (fruit)." The NED algorithm presented here achieves this by generating a new vector embedding on the description of each candidate entity found in the KG, such that the more similar the descriptions, the closer the embeddings of those entities. This is somewhat analogous to what the word-embedding models already do. However, now a full-text document is encoded into a single vector and not just a single word.

Remark that the example of Figure 3 can be presented by an ontology structure. We refer to Related Work section for relevant semantic technologies that can be used to represent service and product names.

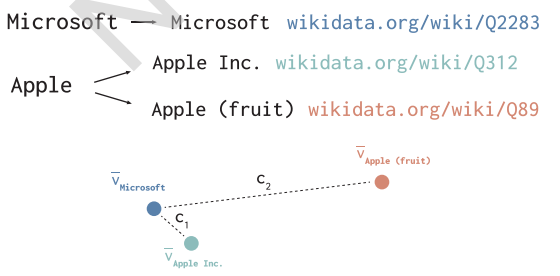

FIG. 3. Context as distance between embedding of entities.

Suppose a list of n similar terms returned by the search-by-similarity feature needs to be disambiguated. The KG is consulted for each of these terms that produce a set of candidate entities per term:

$$Q_i = \text{candidate entities for term } t_i, \text{where } i = 1, \ldots, n$$

then let the set C include all the combinations of candidate entities as follows:

$$C = \{c_j \mid c_j = \langle \vec{v}_1, \ldots, \vec{v}_i, \ldots, \vec{v}_n \rangle \; st \; \vec{v}_i = e(q_i) \text{ and } q_i \in Q_i\}$$

where $e(q_i) = \vec{v}_i$ represents the newly generated document embedding.

The core idea of the algorithm proposed here is to select the combination with the minimum total distance between the entity vectors and the mean vector of the combination itself. Thus the best combination c_{best} is selected in a way such that:

$$c_{best} = \underset{c_j \in C}{\operatorname{argmax}} \sum_{\vec{v}_i \in c_j} cs(m\vec{v}_{c_j}, \vec{v}_i) \quad (1)$$

where $cs(\vec{v}_a, \vec{v}_b)$ is the cosine similarity formula that returns a value $c \in [0, 1]$. The higher c the closer the argument vectors.

Cosine similarity was chosen because in the initial article of Cer et al.,[40] the encoder used to generate the documents embeddings computes similarity utilizing an angular distance built on the cosine similarity. However, due to Equation (1) we find the plain cosine similarity to be already effective.

Equation (1), however, suffers from a major drawback that may occur when the candidate entities of a searched term are very similar to each other. For example, consider the case when "Sprite" and "Pepsi-Cola" need to be disambiguated, as depicted in Figure 4. Suppose that searching "Pepsi-Cola" in the KG produces an entity called "Pepsi-Cola" and another one called "Coca-Cola." Computing the embeddings for these entities gives two possible combinations:

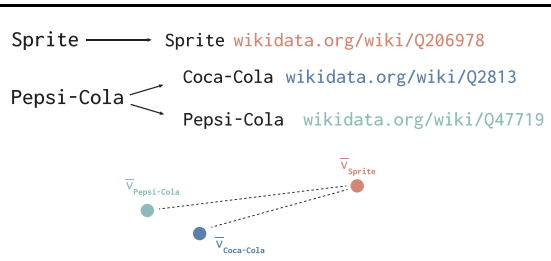

FIG. 4. Failure example of NED based on Equation (1). NED, named-entity–disambiguation.

$$c_1 = \langle \vec{v}_{Sprite}, \vec{v}_{PepsiCola} \rangle \qquad c_2 = \langle \vec{v}_{Sprite}, \vec{v}_{CocaCola} \rangle$$

As both "Pepsi-Cola" and "Coca-Cola" refer to similar carbonated soft drinks, their embeddings will surely be close to each other. However, since both "Coca-Cola" and "Sprite" are produced by The Coca-Cola Company, their embeddings will probably be closer than "Sprite" and "Pepsi-Cola." Hence, Equation (1) would fail, choosing combination c_2 instead of combination c_1.

This issue can be solved by noting that the searched term "Pepsi-Cola" is more similar (equal) to the title of the entity called "Pepsi-Cola" than the title of the entity called "Coca-Cola." For this reason, the Levenshtein-based edit distance ratio $ed(string_1, string_2) \in [0, 1]$ is used between the searched term and the name of each candidate entity. The greater the value of ed the more similar the two words are. Equation (1) now becomes:

$$c_{best} = \underset{c_j \in C}{\operatorname{argmax}} \sum_{\vec{v}_i \in c_j} cs\left(m\vec{v}_{c_j}, \vec{v}_i\right) \cdot ed\left(t_i, title_{q_i}\right) \qquad (2)$$

Unfortunately, the above approach alone fails when the correct candidate entity is not available in the KG. For example, suppose that the NED algorithm processes the words "Apple" and "Microsoft." This time, however, assume that a quick search on the KG for the word "Apple" produces only the entity "Apple (fruit)," as depicted in Figure 5. Equation (2) would choose the only combination available, that is $c = \langle \vec{v}_{\text{Microsoft}}, \vec{v}_{\text{Apple (fruit)}} \rangle$.

This means "Apple (fruit)" is chosen by the algorithm even if it is not the correct entity. In this case, it is preferred to return no entities at all instead of a wrong one. This can be achieved by noting that "Apple (fruit)" is an instance of class "fruit," which in turn is itself an entity in the KG, and since the domain is well

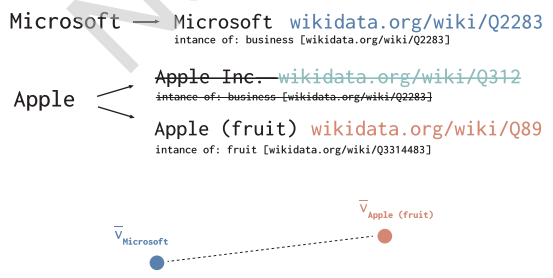

FIG. 5. Failure example of NED based only on Equation (2).

defined, for example, computing products and services, it can be easily rejected. Therefore, once the matched entities are finally chosen using Equation (2), the algorithm performs a final pruning over the entities that do not fall into a fixed set of values for the KG property "instance of." At this point a question arises:

How can we choose a complete set of values for this property, to have full coverage of the products, services, and vendors in the KG?

Sadly, every entity in the KG can be used as a value for this property; therefore, the number of potential classes to choose from can be potentially extremely large, thus making it nearly impossible to define a complete set of classes of interest.

In this work, this problem is addressed in two steps. First, a small set of entries from the KG are manually flagged, among the most common "instance of" entities that appeared in the test set described in Data Sources section. Then, the algorithm uses a nearest neighboring approach trained over the space of the embeddings of all the candidate entities. Every matched entity with property "instance of" outside this fixed set is preserved only whether enough of their nearest neighbors have their value within it.

Case Study in the Financial Sector

To provide a better understanding of the components of the methodology and how they operate in practical scenarios, an application in the financial sector is implemented that is inspired by the Italian central bank, that is, the Bank of Italy. A high-level presentation of the services that comprise the resulting system is depicted in Figure 6.

As defined by the generic methodology presented in Generic Methodology for Supplier Discovery section, in the first step data are collected from Reddit to form the basis of the knowledge base of the system. Then, the text is encoded in a way such that similarity searches can be performed for particular products or vendors, by training a word-embedding model over the dataset. For the disambiguation of the results obtained, Wikidata is used by the NED algorithm. Finally, a query processor is implemented to allow the user to carry out procurement queries inferencing the trained model and passing the output to the NED algorithm. Remark that after the initial execution of the four phases of the methodology, the first two phases can continue their operation independently from the query processor, continuously monitoring the identified social media for new posts and incrementally

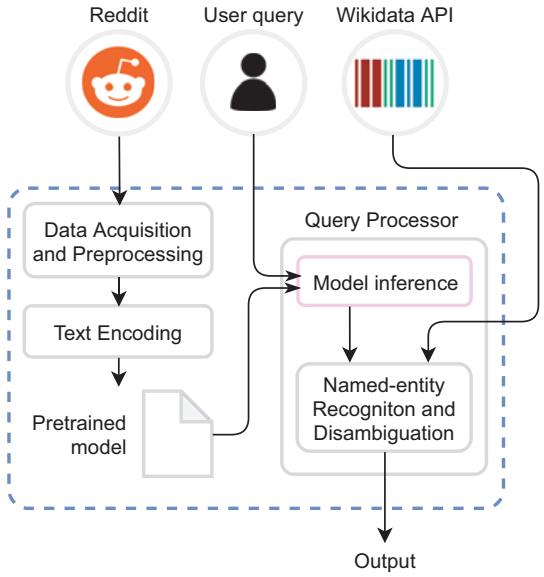

FIG. 6. High-level interaction of the main components that make up the implemented system for supplier discovery.

updating the NLP models. Selected technical aspects of the system components are presented in the following sections.

Data sources

For the needs of the specific case study, the Reddit social news aggregation, web content rating, and discussion website is selected as the primary source of raw data. With a focus on the financial services market and in particular on financial information services, a total of 66 topics are identified, that is, subreddits, that include posts in the English language. Once the pertinent subreddits have been manually selected, data are collected through the Pushshift API, providing an extract of all comments and posts from January 2016 to February 2020. In total more than 2.3M posts were retrieved for a total of 36 GB. The data were preprocessed by removing any markdown syntax, to extract the plain text of submissions and comments retrieved. The POS-fast model of the Flair library[§§] was selected as it implements state-of-the-art NLP techniques.

[§§]https://github.com/flairNLP/flair/blob/master/resources/docs/TUTORIAL_2_TAGGING.md

Table 2. Word-embedding model trained with different hyperparameters

Model name	*min_count*	*vector_size*	*lm*	*Epochs*
m1_s300_cb_5	1	300	cb	5
m1_s100_cb_5	1	100	cb	5
m5_s100_cb_5	5	100	cb	5
m5_s100_sg_5	5	100	sg	5
m5_s300_cb_5	5	300	cb	5
m5_s300_sg_5	5	300	sg	5

Text encoding

For the text encoding phase, multiple implementations of different word-embedding models were developed to allow a detailed evaluation of the achieved performance. All the implementation of the word embeddings was based on the Google Word2Vec (w2v) model.[41,42] The specific model was used after carrying out a preliminary analysis using the GoogleNews Word2Vec pretrained model[***] that indicates that it is not affected by disambiguation cases where part of a vendor/product name has also meaning on its own.

There are of course other word-embedding models that could be used as a part of Word2Vec, like GloVe or fastText. While embeddings generated by Word2Vec and GloVe tend to perform very similarly,[†††] fastText can also handle words that never appear in the training dataset (OOV, out-of-vocabulary) by splitting each word in smaller n-grams. However, if we compare the results of Word2Vec against fastText, as done in Rathore,[43] we can see that the Word2Vec model seems to perform better on semantic tasks because the information from irrelevant n-grams worsens the embeddings. In contrast, fastText embeddings are significantly better at encoding syntactic information, which however is not needed in our case.

Multiple different word-embedding model instances were trained using Word2Vec implemented by Gensim library,[‡‡‡] which is an open-source library for unsupervised topic modeling and NLP, tuned with different values for the following hyperparameters.[§§§] The resulting models have been evaluated using different performance criteria as presented in Performance Evaluation section. Table 2 lists six models that achieved the best results among those tested. The names of the models are encoded as follows:

[***]https://code.google.com/archive/p/word2vec
[†††]https://www.quora.com/How-is-GloVe-different-from-word2vec
[‡‡‡]https://radimrehurek.com/gensim_3.8.3
[§§§]https://radimrehurek.com/gensim/models/word2vec.html#gensim.models.word2vec.Word2Vec

- `min_count` $\in [5, 25]$—minimum frequency of the word in the entire text corpus required to be considered in the creation of the statistical model.
- `vector_size` $\in [100, 300]$—the dimensions of the word vectors used to associate each word in the vocabulary a distributed "feature vector," thereby creating a notion of similarity between words.
- `lm` $\in \{$`sg`, `cb`$\}$—the learning models used by word2vec: continuous Skip-Gram (1) or Continuous Bag-of-Words, or CBOW model (2). Both models are focused on learning about words given their local usage context, where the context is defined by a window of neighboring words. The continuous skip-gram model learns by predicting the surrounding words given a current word. The CBOW model learns the embedding by predicting the current word based on its context.
- `epochs` $\in [5, 15]$—number of training iterations for creating the statistical model.

Choice of the KG

Given that the application is required to discover vendors and products, Wikidata is preferred over other open KGs, because of its bigger number of entities.[36] Each entity in Wikidata represents a topic, concept, or object and has its identifier. Statements are recorded as property-value pairs. In this work, we are mainly interested in:

(1) Property "official website": The official website of the entity, if available.
(2) Property "instance of": That class of which the current entity is a particular example and member.
(3) Entity description and Wikipedia page link.

Wikidata and Wikipedia are accessed through the official API at runtime that also provides the HTML meta-description tag of the official website page linked in the Wikidata entities.

Named-entity–disambiguation

First, it is important to clarify some differences between applying NED on a list of similar terms returned by Word2Vec trained over the Reddit dataset instead of on a sentence. Assume for example to perform a similarity search of the word "bloomberg_terminal" into model `m5_s300_cb` which, as indicated in the performance evaluation presented in Performance Evaluation section, is the model that achieves the highest accuracy among all the trained models. The 15 most similar terms for this word, together with their cosine similarity, are the following:

- bloomberg_terminals [0.749]
- capital_iq [0.686]
- datastream [0.667]
- capiq [0.665]
- eikon [0.646]
- wrds [0.648]
- capitaliq [0.612]
- yahoo finance [0.613]
- bloomberg [0.615]
- factset [0.606]
- quandl [0.607]
- bamsec [0.583]
- ycharts [0.582]
- reuters eikon [0.576]

A quick examination of the above list points out some particularities. First, the same product may appear multiple times with slightly different names or typing errors, for example, "capital_iq" and "capitaliq." Second, the similarity score provided by the word-embedding model cannot be trusted too much in detail. Consider, for example, "reuters_eikon" which is the same product as "eikon," but it is at the bottom of the list with the lowest similarity. To overcome these shortcomings, the NED algorithm is extended by incorporating a preliminary check on misspelt input words. If a pair of very similar words has been detected, using a Levenshtein-distance–based formula,[****] the algorithm preserves only the one with the highest number of occurrences in the Reddit dataset.

Remark that due to the high dimensionality of the embeddings generated in the text encoding step when the USE is applied, the final pruning over the disambiguated entities uses an Approximated Nearest Neighboring approach for fast classification, implemented using the Spotify Annoy library.[††††]

Training time

The experiments reported here are based on an i7-4770 CPU with 16 GB RAM and an Nvidia GTX 750ti GPU over an Ubuntu 18.04 OS, python 3.7, and gensim 4.2.0. The training time of the word embedding models is provided in Table 3, together with their size and initialization time. Considering that these models will

[****] https://en.wikipedia.org/wiki/Levenshtein_distance
[††††] https://github.com/spotify/annoy

Table 3. Total size, training, and loading time of different word-embedding models

Model name	*Size*	*Training time*	*Init. time*
m1_s300_cb_5	8.5 GB	2 Days 9 hours 12 minutes	3 Minutes 57 seconds
m1_s100_cb_5	2.92 GB	2 Days 4 hours 50 minutes	2 Minutes 57 seconds
m5_s100_cb_5	338 MB	2 Days 3 hours 44 minutes	18 Seconds
m5_s100_sg_5	338 MB	2 Days 4 hours 35 minutes	16 Seconds
m5_s300_cb_5	993 MB	2 Days 8 hours 25 minutes	20 Seconds
m5_s300_sg_5	993 MB	2 Days 9 hours 14 minutes	22 Seconds

need to be loaded in-memory, the size of the models is important to understand the hardware RAM requirements needed at runtime.

The query time needed by the web application to display the results, which involves either querying the word-embedding model or computing the NED algorithm, ranges on average between 5 and 15 seconds.

Query processor and postanalysis

The system developed is accessed through a reactive web application with the appearance of a search engine. The application is built with a Javascript front-end framework called Svelte and a Python back-end framework called AIOHTTP. Particular attention has been paid to UX and UI design, to provide a pleasant and fast user experience.

The GUI features a large search bar that can be used to input the name of a financial service the user wants to find alternatives. As soon as the search button is pressed, the whole machine-learning pipeline gets triggered. The web server inferences the pretrained word-embedding model to get the most semantically similar words to be passed to the NED algorithm, which asynchronously calls the Wikidata API to find pertinent entities. The design of the GUI aims to make the consultation of such results straightforward. In Figure 7 we provide a screenshot of the interface, with the addition of some dashed areas that we are going to use to better explain the functionalities of the application.

First, the query term is presented to the user along with some extra information found on the KG (Area A). Second, all the similar financial services that are identified by the algorithm are displayed in the form of a list (Area B) along with additional information such as:

(1) The similarity score is computed using the cosine similarity of USE embedding.
(2) The number of occurrences for the word in the data sources.
(3) A short description, the vendor's official website link, and the Wikipedia page link retrieved from the KG.

These entities are visualized using a force graph (Area E) as red nodes, while the query term shows up with a yellow color. This graph shows the interconnections between nodes, and the force between the nodes is set to the similarity score provided by the model. It allows one to return to the search page of a result, just by clicking the corresponding node.

The graph includes some additional gray-colored nodes. These nodes represent semantically similar words that were not matched against any entity by the NED algorithm as no entry was available in the KG. It is expected that these nodes may provide valuable insights into very recent products and services. Therefore, they are also listed in Area C, with shortcut buttons to search them in the Google search engine or to create a new item in the KG.

Finally, all the words pruned at some point by the NED algorithm are listed in Area D. Remark that as it will be evaluated in the next section, with a high probability these words are of no interest as they are not connected to any relevant product or service.

The application implemented is evaluated in detail in Evaluating the Effectiveness section, and evidence is provided on how the system allowed the Bank of Italy to discover multiple new products in the specific sector of interest that were previously unknown.

Performance Evaluation

In this section the performance of the resulting application of the methodology in the Financial Service sector is evaluated using quantitative and qualitative criteria. The accuracy of the text encoding phase and specifically the identification of entity names that are relevant is evaluated using a labeled dataset created with the assistance of a group of business experts from the Bank of Italy.

Accuracy of text encoding

The text encoding phase is an unsupervised learning problem that creates a model that is used to identify similar entities in the data sources. Developing accurate models for named entity recognitions requires large

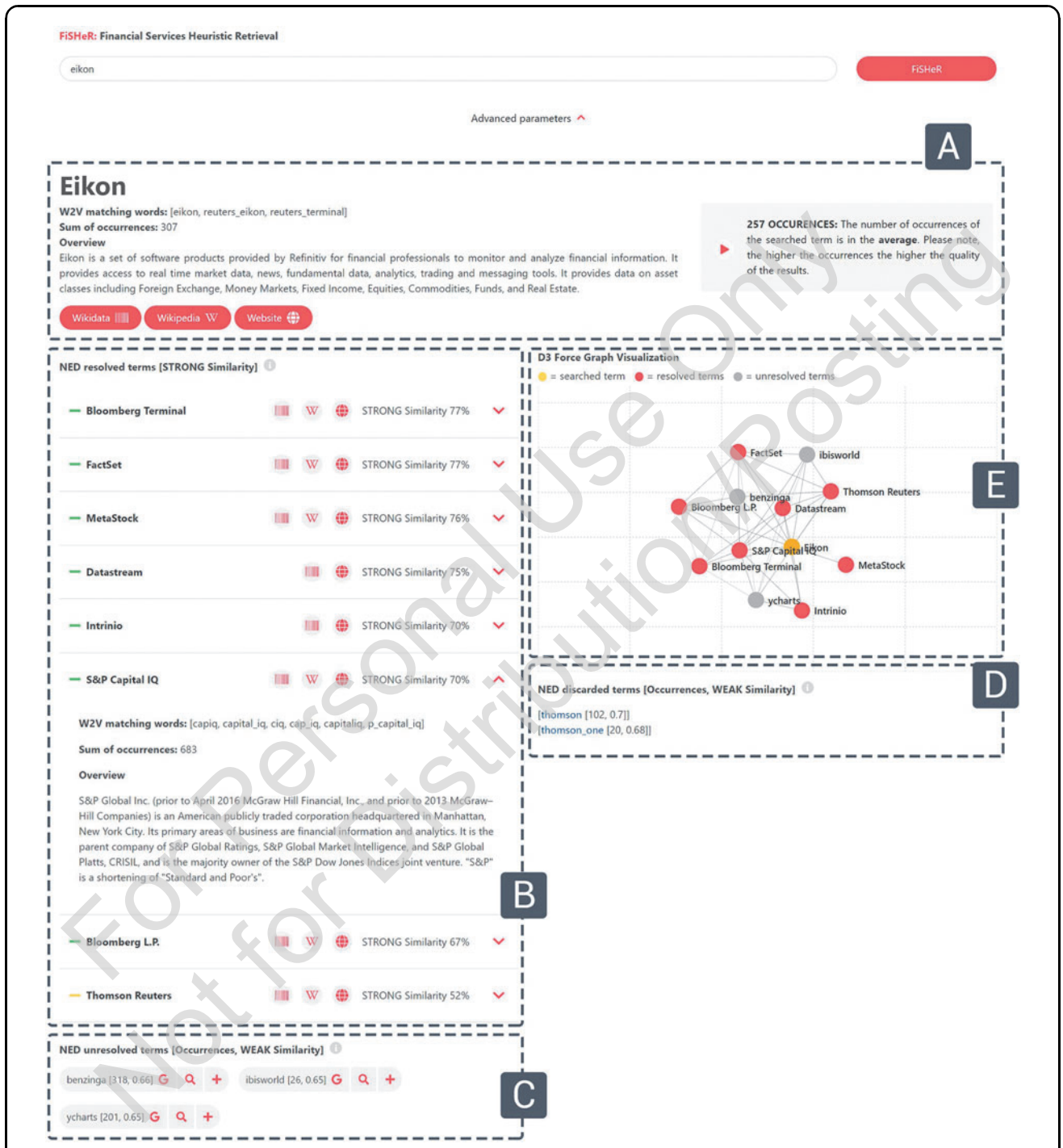

FIG. 7. Screenshot of the web application that displays the query results split into five logical areas. Area A contains information on the searched term retrieved from the knowledge graph, that is, from Wikipedia; Area B contains pertinent results together with Wikidata information; Area C lists pertinent results not found in the KG. Area D displays results flagged as irrelevant; Area E provides a graph with pertinent results, both found and not-found in the KG. See section "Case Study in the Financial Sector", subsection "Query processor and postanalysis" for the description of each area for the actual results extracted from the data collected from Reddit that were posted from January 2016 to February 2020. KG, knowledge graph.

Table 4. Names of existing products and services in the financial sector identified by a group of business experts

Query terms	
bloomberg_terminal	Finastra
dow_jones	Iboxx
eikon	Intex
euro_stoxx	Metastock
morningstar	Pitchbook
morningstar_ratings	Refinitiv
morningstar_research	reuters_news
msci_emerging_markets	Stoxx
moodys_analytic	Tradeweb
vanguard_money_market	Xetra
factset	Ycharts

amounts of knowledge in the form of feature engineering and lexicons as discussed in Related Work section.

Due to the lack of such large corpus of training data, in this section we follow a different approach for the evaluation of the performance. In particular we consult a group of business experts to define a set of 22 financial products that will be used to query the resulting model. Table 4 depicts this list of financial products. For each of these product names, the model is queried, and the top-15 most similar terms are recorded. In total $15 \times 22 \times 6 = 1980$ were produced out of which the unique results are 966. In the sequel, the complete list of 966 identified entities that the model indicated as similar is given to the group of business experts for evaluation. Each entry is evaluated as "GOOD" if the result is considered a product relevant to the one used in the query, or "BAD" if it is not. Table 5 shows an example of the output of this evaluation.

The resulting terms, along with the evaluation of the expert, were used to build a labeled dataset for the evaluation of the word-embedding models. Based on this dataset, the performance of a model used for the text-encoding phase can be evaluated by assessing the achieved *accuracy* as follows:

Table 5. Example of an evaluation conducted over some of the results produced by the query term "morningstar" for word-embedding model m5_s300_cb

Query term	*Result term*	*w2v cosine sim.*	*Label*
morningstar	morning_star	0.834	GOOD
morningstar	yahoo_finance	0.769	GOOD
morningstar	trustnet	0.761	GOOD
⋮	⋮	⋮	⋮
morningstar	vro	0.660	BAD
morningstar	canstar	0.659	GOOD
morningstar	value_research	0.654	GOOD

For the explanation of "w2v cosine sim.," see Named-Entity Recognition and Disambiguation section.

w2v, Word2Vec.

$$Accuracy = \frac{\text{Number of results labeled as GOOD}}{\text{Total number of results}}. \quad (3)$$

Table 6 summarizes the performance of six models that achieved the highest accuracy.

The performance evaluation indicates that the model `m5_s300_cb_5` achieves the best accuracy with 80% of good results and model `m1_s300_cb_5` follows with 77%. The results indicate that the approach of skipping words with several occurrences that are too low, while training the word-embedding model, seems to slightly increase the accuracy for this test set.

Remark that the models `m5_s300_cb_5` and `m5_s300_sg_5` are trained with the same values for `vector_size` and `min_count`. In other words, for the specific dataset and evaluation methodology, the evaluation results indicate that the models constructed using the CBOW learning model reach a higher accuracy than those built using the Skip-Gram model.

Accuracy of NED

In this section, the results acquired from the text encoding phase are examined using the NED algorithm to evaluate the accuracy in terms of identifying relevant products and services. The evaluation of the entities provided by the NED algorithm is conducted with the support of a group of business experts so that false results are identified.

Recall from Named-Entity Recognition and Disambiguation section that the NED algorithm is not in a position to resolve a term whose corresponding entity is missing from the KG. Thus the terms that are not present in the KG are excluded from the evaluation of the accuracy of the algorithm. As a result, the number of samples in the test set, defined in Accuracy of Text Encoding section, drops from 966 to 590 elements.

Table 6. Accuracy for the word-embedding models that achieved the highest accuracy

Model name	*Accuracy*
m1_s300_cb_5	0.778
m1_s100_cb_5	0.715
m5_s100_cb_5	0.757
m5_s100_sg_5	0.712
m5_s300_cb_5	**0.800**
m5_s300_sg_5	0.739

Best performance is displayed in bold.

Table 7. Definition of true positives, true negatives, false positives, false negatives, looking at the entity found by the named-entity–disambiguation algorithm for each identified similar term during the text encoding based on the manual evaluation by the group of business experts

Label	*NED returns an entity?*	*Correct?*	*Treated as*
GOOD	Yes, $\geq$min_sim	Yes	TP
GOOD	Yes, $\geq$min_sim	No	FP
BAD	Yes, $\geq$min_sim	Yes	FP
BAD	Yes, $\geq$min_sim	No	FP
GOOD	No	No	FN
BAD	No	No	TN
GOOD	Yes, $<$min_sim	Yes	FN
GOOD	Yes, $<$min_sim	No	FN
BAD	Yes, $<$min_sim	No	FN
BAD	Yes, $<$min_sim	Yes	TN

FN, false negatives; FP, false positives; TN, true negatives; TP, true positives.

The resulting dataset of 590 elements was examined by the group of business experts to validate if they are indeed existing product/services or not. Since the desired output of the NED algorithm is a financial product or vendor, a result is classified as incorrect if it does not correspond to a financial product/vendor, even if it is a generic financial term.

The NED algorithm is capable of discarding results that are not relevant through filtering the KG. This is the pruning step described at the end of Named-Entity Recognition and Disambiguation section. In the specific application where Wikidata is used as the KG, this is done using the "instance of" property of Wikidata. Given the results acquired from looking up the KG, the algorithm uses a nearest neighbor search algorithm to select the results that are most similar to the product/service name used in the query. The number of results that the nearest neighbor search algorithm will use during the training cannot be fixed as it depends on the candidate entities found at runtime. Therefore, a fixed number of the most relevant candidate entities were retrieved and then we filtered out those below a minimum threshold of similarity.

Recall that the similarity between results is computed using the cosine similarity score. The number of nearest neighbors, k, the minimum similarity, `min_sim`, and the minimum threshold of neighbors, `min_nn`, are considered hyperparameters that need to be evaluated while computing accuracy measurements. In terms of $k=9$ different values of even numbers were tested, while for `min_nn`, since we want the number of good surrounding entities to be greater than the bad ones, the threshold values tested were in $\{\frac{6}{9}, \frac{7}{9}, \frac{8}{9}, \frac{9}{9}\} = \{0.65, 0.75, 0.85, 1\}$.

Given the above hyperparameters, the dataset labeled by the group of business experts and the results produced by the NED algorithms are classified with the help of the group of business experts as True Positives (TP), True Negatives (TN), False Positives (FP), or False Negatives (FN) using a simple decision tree process which is summarized in Table 7. Remark that the results that lack a corresponding entity in the KG are completely excluded from the evaluation process.

The performance of the NED algorithm is based on the accuracy, f1 score, recall, and precision achieved which are calculated as follows:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

Figure 8 includes the results for three instances that achieve the best results. In particular, the maximum accuracy achieved is 0.766 with `k=9`, `min_nn=1`, `min_sim=0.50`. The highest F1 score is 0.821 with `k=9`, `min_nn=1`, `min_sim=0.50`.

Evaluating the effectiveness

In this section, we wish to evaluate the effectiveness of the developed application using qualitative terms. Our approach is to allow the group of business experts from the Bank of Italy to use the web application during the process of discovering new products and services. The number of discovered services depends on the business experts' previous knowledge and it does not capture directly in a quantitative way the accuracy of the system, yet it provides an indication of the effectiveness of the methodology in a real-world setting.

Table 8 summarizes the findings of this evaluation. The number of discovered services may overlap among different text encoding models. As we can see from the second column, which represents the unknown new products, `m5_s100_sg_5` has the highest number of discovered services. Initially, we thought this could be related to the usage of Skip-gram, as "Skip-gram works well with small amount of

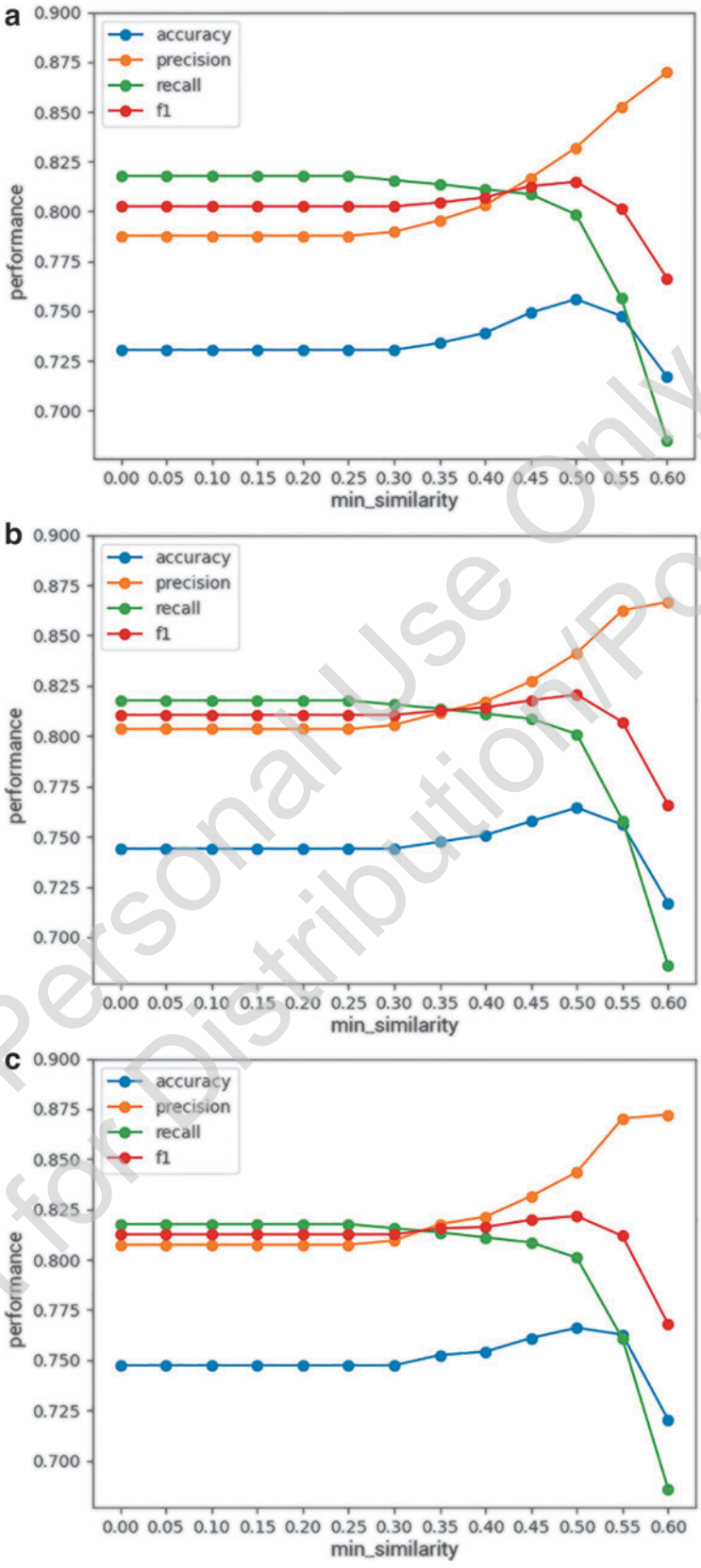

FIG. 8. Performance achieved by the NED algorithm for a selected number of k and `min_nn`. The *x*-axis indicates the `min_sim` hyperparameter variable, while the *y*-axis provides performance values. **(a)** $k = 9$, min_nn = 0.75. **(b)** $k = 9$, min_nn = 0.85. **(c)** $k = 9$, min_nn = 1.

Table 8. Number of newly discovered financial services out of 330 results and percentages (second column); accuracy of the Wikidata entities matched and not pruned by the named-entity–disambiguation algorithm (third column)

Model name	*Disc. services*	*Accuracy of NED*
m1_s300_cb_5	26 (7.8%)	0.927 (183 Results left)
m1_s100_cb_5	32 (9.6%)	0.880 (150 Results left)
m5_s100_cb_5	33 (10.0%)	0.867 (171 Results left)
m5_s100_sg_5	40 (12.3%)	0.935 (128 Results left)
m5_s300_cb_5	26 (7.8%)	0.940 (169 Results left)
m5_s300_sg_5	29 (8.7%)	0.910 (137 Results left)

the training data, represents well even rare words or phrases; CBOW is several times faster to train than the skip-gram, slightly better accuracy for the frequent words."[‡‡‡‡] However, we would have expected to see a similar spike also in the model `m5_s300_sg_5` which was trained with Skip-gram too, but this did not happen. Therefore, we do not have any strong statistical evidence that Skip-gram performs better to find less-known financial products in this test set.

Moreover, if we want to compute the accuracy of area B of the web application user interface (Query Processor and Post Analysis section), which displays only the matched entities in the KG, we can think about it as the accuracy of the word-embedding model (Text Encoding section), removing from the test set all the unresolvable terms because of missing corresponding entity in the KG, and all the terms not matched by the NED algorithm or with a similarity lower than `min_sim`.

For this evaluation, we set the hyperparameters `k=9`, `min_nn=1`, and `min_sim=0.50`, following the findings of Accuracy of NED section. Note that the NED algorithm may return slightly different results at every computation. Therefore, we repeated the computation thrice, and we report the mean values in the third column of Table 8. Again, model `m5_s300_cb_5` reaches the best accuracy followed by `m5_s100_sg_5` and `m1_s300_cb_5`. Interestingly, `m5_s100_sg_5` has a higher accuracy increment, if compared to the values in Table 6, than the other models. However, it has only 128 samples versus the 169 of model `m5_s300_cb_5` and the 183 of model `m1_s300_cb_5`.

Implications on Tender Procedures and Limitations

The tool presented here does not provide any guarantees that a complete view of the financial market segment will be created. Given that the starting point of the methodology presented here is the identified social media, it is reasonable to accept that there might exist some financial products that are never mentioned in the posts and corresponding discussions. Probably such financial products are very recently introduced by newly founded start-ups. In this way, since these products are not mentioned in the social media, they will not be part of the trained models and therefore when the user is querying the tool, they will not be included in the results.

One way to mitigate this risk is by first scrapping data from multiple social media. A second way is to continually monitor these social media so that when at some point in the future a new post mentions these products, it will be retrieved and eventually the product names will be included in the resulting models. Finally, a third approach is to publicly announce the social media monitored for the construction of the models of the supplier discovery tool so that newly funded start-ups can proactively mention their products in one of these social media.

It is also important to consider that certain financial products that are produced by newly founded start-ups may be erroneously removed from the KG during the named entity recognition and disambiguation phase. Recall that the method relies on Wikidata to disambiguate supplier names. If a newly founded company does not have an entry in Wikidata, it will not be identified as a valid supplier.

One way to mitigate this is by inviting all newly introduced suppliers to create an entry in Wikidata describing their company. Another approach is to extend the system to also include public registries for company data, such as the Global Legal Entity Identification System,[§§§§] the European Business Register (EBR),[*****] the Dun & Bradstreet DUNS,[†††††] or the Business Register Exchange (BREX).[‡‡‡‡‡] The data provided in these registries can be further enriched through specific ontologies that capture company and company-related data, such as the W3C Organization

[‡‡‡‡]https://groups.google.com/g/word2vec-toolkit/c/NLvYXU99cAM/m/E5Id8LcDxIAJ

[§§§§]https://www.gleif.org
[*****]https://ebra.be
[†††††]https://www.dnb.com/duns.html
[‡‡‡‡‡]https://www.kompany.com

ontology (ORG),[§§§§§] the e-Government Core Vocabularies,[******] or the Financial Industry Business Ontology (FIBO).[44] More recently, in Roman et al.,[45] the euBusinessGraph ontology is introduced in an attempt to support tools like the one presented here to improve the reconciliation process matching supplier data against company data. Although none of these registries and ontologies provide complete coverage of cross-border and cross-language company data, using them in combination may help further reduce the chances of not rejecting a valid supplier during the disambiguation phase.

Given the above limitations, the generic methodology presented here and the implemented tool can facilitate the applicability of principles of transparency, proportionality, efficiency, and obligation-to-state in the public procurement process. In particular, it allows to:

(1) Increase the pool of potential providers known in the market;
(2) Highlight possible savings related to the selection of lower-cost services from a greater number of market players than hitherto considered;
(3) Help identify the correct procedure for selection and acquisition of services on the market.

Indeed, the financial services market is inherently uneven, with some segments characterized by multiple competing players and other segments with monopolist providers. In the latter case, it is generally possible for a European institution to use a negotiated tendering without prior publication of a contract notice. Citing from the Directive 2014/24/EU of the European Parliament and of the Council of 26 February 2014 on public procurement:

> This exception should be limited to cases where […] it is clear from the outset that publication would not trigger more competition or better procurement outcomes, not least because there is objectively only one economic operator that can perform the contract.

Therefore, the machine-learning methodology presented in this article can be an important auxiliary tool that, leveraging data sources never used before, can either (1) further substantiate a hypothesis that a particular single supplier is not fungible or (2) completely dismantle it by identifying alternative solutions that are nevertheless capable of meeting the needs of the institution.

Conclusions

The work presented here focuses on the supplier-discovery part of the procurement process, supporting market-research and helping to overcome disadvantages in terms of resources and capabilities that hinder innovative SME participation in public procurements. A novel methodology is presented that passively monitors publicly available conversations conducted in focused discussion groups in social media to identify data sources that can provide valuable insights about innovative and emerging products and services. A carefully designed combination of NLP techniques processes the data extracted from the social media, identifies potential products and services based on KGs available on the internet, and evaluates the similarity of the identified offerings. Remark that the proposed methodology does not explicitly rely on networks of peers and does not use historical open procurement datasets to identify suppliers.

We apply the methodology on a very specific use case in the financial sector based on the needs of the Bank of Italy. We develop an interactive web-based application environment that allows to query the NLP-based models to explore the extracted knowledge. The resulting market-research tool starts from a specific already known product and uses the extracted knowledge base to explore similar products that were previous unknown.

We evaluate the performance of the developed application both in terms of quantitative and qualitative criteria with the support of a group of experts from the Bank of Italy. Our evaluation demonstrates the benefits of extracting knowledge passively from data that have already been published autonomously on the internet and highlights the capability of the proposed methodology to identify data sources from which it is possible to acquire information that can potentially be of value in the process of supplier discovery. We believe that the results presented here can provide valuable indications to address the specific needs identified in Muninger et al.[13] on how to efficiently and effectively integrate technological advancements and AI techniques in the process of discovering suppliers.

A possible future work direction includes the enrichment of the dataset constructed here so that it can become sufficiently large to support the training of deep

[§§§§§] https://www.w3.org/TR/vocab-org
[******] https://joinup.ec.europa.eu/solution/e-government-core-vocabularies

neural network models presented in Related Work section. It is our goal to evaluate a selected number of state-of-the-art algorithms for NED and comparatively study the resulting end-to-end performance.

Authors' Contributions

M.P.: conceptualization (equal); methodology (equal); software (lead); experiments (lead); writing—original draft (equal). I.C.: conceptualization (equal); methodology (equal); writing—original draft (equal); writing—review and editing (lead). A.A.: conceptualization (equal); methodology (equal); writing—original draft (equal); writing—review and editing (supporting).

Disclaimer

The views expressed in the articles are those of the authors and do not involve the responsibility of Bank of Italy.

Author Disclosure Statement

No competing financial interests exist.

Funding Information

Supported by the ERC Advanced Grant 788893 AMDROMA, EC H2020RIA project "SoBigData++" (871042), PNRR MUR project PE0000013-FAIR, PNRR MUR project IR0000013-SoBigData.it.

References

1. Organisation for Economic Co-operation and Development (OECD). Government at a Glance 2015. OECD Publications: Berlin, Germany; 2015.
2. Organisation for Economic Co-operation and Development (OECD). Productivity in Public Procurement: A Case Study of Finland: Measuring the Efficiency and Effectiveness of Public Procurement. Technical Report. OECD; OECD Publications: Berlin, Germany; 2019.
3. Brammer S, Walker H. Sustainable procurement in the public sector: an international comparative study. Int J Oper Prod Manage 2011;31(4): 452–476.
4. Pro Inno Europe. Guide on Dealing with Innovative Solutions in Public Procurement: 10 Elements of Good Practice. EU Publications: Luxembourg; 2007.
5. Georghiou L, Edler J, Uyarra E, et al. Policy instruments for public procurement of innovation: Choice, design and assessment. Technol Forecast Soc Change 2014;86:1–12; doi: 10.1016/j.techfore.2013.09.018
6. Organisation for Economic Co-operation and Development (OECD). OECD Economic Outlook. Technical Report. Volume 2016, Issue 1. OECD; OECD Library; 2016.
7. Edquist C, Hommen L. Systems of innovation: Theory and policy for the demand side1this article is based on work from the project "Innovation Systems and European Integration (ISE)," funded by Targeted Socio-Economic Research, DG XII, European Commission, Contract No. SOE1-CT95-1004 (DG 12-SOLS). In particular, the article draws upon work originally produced as part of ISE subproject 3.2.2, "Public Technology Procurement as an Innovation Policy Instrument." Technol Soc 1999; 21(1):63–79; doi: 10.1016/S0160-791X(98)00037-2
8. Baltrunaite A, Giorgiantonio C, Mocetti S, et al. Discretion and supplier selection in public procurement. J Law Econ Organization 2021;37(1): 134–166.
9. Smith C. The wholesale and retail markets of London, 1660–1840. Econ Hist Rev 2002;55(1):31–50.
10. Howells J. Intermediation and the role of intermediaries in innovation. Res Policy 2006;35(5):715–728; doi: 10.1016/j.respol.2006.03.005
11. Nambisan S, Lyytinen K, Majchrzak A, et al. Digital innovation management: Reinventing innovation management research in a digital world. MIS Q 2017;41(1):223–238.
12. Obwegeser N, Müller SD. Innovation and public procurement: Terminology, concepts, and applications. Technovation 2018;74–75:1–17; doi: 10.1016/j.technovation.2018.02.015
13. Muninger M-I, Mahr D, Hammedi W. Social media use: A review of innovation management practices. J Bus Res 2022;143:140–156; doi: 10.1016/j.jbusres.2022.01.039
14. Lewis JA, Odeyinka H, Eadie R. Innovative construction procurement selection through an artificial intelligence approach. Management 2011;745:754.
15. Futia G, Morando F, Melandri A, et al. Contrattipubblici.org, a Semantic Knowledge Graph on Public Procurement Information. In: AI Approaches to the Complexity of Legal Systems. (Pagallo U, Palmirani M, Casanovas P, et al. eds.) Springer: Midtown Manhattan, New York City; 2015; pp. 380–393.
16. Aravena-Díaz V, Gacitúa R, Astudillo H, et al. Identifying Potential Suppliers for Competitive Bidding Using Latent Semantic Analysis. In: 2016 XLII Latin American Computing Conference (CLEI). IEEE: Manhattan, New York; 2016; pp. 1–12.
17. Haanpää A. Applying Natural Language Processing in Text Based Supplier Discovery. Master's Thesis, Tampere University: Tampere, Finland; 2019.
18. Yaozu Y, Jiangen Z. Constructing government procurement knowledge graph based on crawler data. J Phys Conf Ser 2020;1693(1):012032; doi: 10.1088/1742-6596/1693/1/012032
19. Soylu A, Corcho O, Elvesæter B, et al. Enhancing Public Procurement in the European Union Through Constructing and Exploiting an Integrated Knowledge Graph. In: International Semantic Web Conference. (Pan JZ, Tamma V, d'Amato C, et al. eds.) Springer: Midtown Manhattan, New York City; 2020; pp. 430–446.
20. Soylu A, Corcho O, Elvesæter B, et al. TheyBuyForYou platform and knowledge graph: Expanding horizons in public procurement with open linked data. Semantic Web 2022;13(2):265–291; doi: 10.3233/SW-210442
21. Soylu A, Corcho Ó, Elvesæter B, et al. Data quality barriers for transparency in public procurement. Information 2022;13(2):99; doi: 10.3390/info13020099
22. Guasch C, Lodi G, Van Dooren S. Semantic Knowledge Graphs for Distributed Data Spaces: The Public Procurement Pilot Experience. In: The Semantic Web—ISWC 2022, Lecture Notes in Computer Science. (Sattler U, Hogan A, Keet M, et al. eds.) Springer International Publishing: Midtown Manhattan, New York City; 2022; pp. 753–769; doi: 10.1007/978-3-031-19433-7_43
23. Benítez-Hidalgo A, Barba-González C, García-Nieto J, et al. TITAN: A knowledge-based platform for big data workflow management. Knowl Based Syst 2021;232:107489; doi: 10.1016/j.knosys.2021.107489
24. Barlaug N, Gulla JA. Neural networks for entity matching: A survey. ACM Trans Knowl Discov Data 2021;15(3):1–37.
25. Parravicini A, Patra R, Bartolini DB, et al. Fast and Accurate Entity Linking via Graph Embedding. In: Proceedings of the 2nd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA). Amsterdam, Netherlands; 2019; pp. 1–9.
26. Lin JC-W, Shao Y, Zhou Y, et al. A Bi-LSTM mention hypergraph model with encoding schema for mention extraction. Eng Appl Artif Intell 2019;85:175–181.
27. Lin JC-W, Shao Y, Zhang J, et al. Enhanced sequence labeling based on latent variable conditional random fields. Neurocomputing 2020;403: 431–440.
28. Shao Y, Lin JC-W, Srivastava G, et al. Self-attention-based conditional random fields latent variables model for sequence labeling. Pattern Recognit Lett 2021;145:157–164.
29. Lin JC-W, Shao Y, Djenouri Y, et al. Asrnn: A recurrent neural network with an attention model for sequence labeling. Knowl Based Syst 2021;212:106548.
30. Engel Y, van Werven R, Keizer A. How novice and experienced entrepreneurs name new ventures. J Small Bus Manage 2022;60(4):828–858.
31. Stoyneva I, Vracheva V. Demystifying entrepreneurial name choice: Insights from the us biotech industry. N Engl J Entrepreneurship 2022; 25(2):121–143.

32. Belz A, Graddy-Reed A, Shweta FNU, et al. Deterministic Bibliometric Disambiguation Challenges in Company Names. In: 2023 IEEE 17th International Conference on Semantic Computing (ICSC). Laguna Hills, CA, USA; 2023; pp. 239–243; doi: 10.1109/ICSC56153.2023.00047
33. Belz A, Graddy-Reed A, Shweta FNU, et al. Patentopia: A multi-stage patent extraction platform with disambiguation for certain semantic challenges. IEEE International Conference on Big Data (Big Data): Osaka, Japan; 2022; pp. 3478–3485; doi: 10.1109/BigData55660.2022.10020918
34. Primpeli A, Peeters R, Bizer C. The WDC Training Dataset and Gold Standard for Large-Scale Product Matching. In: Companion Proceedings of the 2019 World Wide Web Conference, WWW'19. Association for Computing Machinery: New York, NY, 2019; pp. 381–386; doi: 10.1145/3308560.3316609
35. Basile A, Crupi R, Grasso M, et al. Disambiguation of company names via deep recurrent networks. arXiv preprint arXiv:2303.05391, 2023.
36. Färber M, Rettinger A. Which knowledge graph is best for me? arXiv preprint arXiv:1809.11099, 2018.
37. Horan C. When Not to Choose the Best NLP Model. 2019. Available from: https://blog.floydhub.com/when-the-best-nlp-model-is-not-the-best-choice [Last accessed: July 3, 2023].
38. Baumgartner J, Zannettou S, Keegan B, et al. The Pushshift Reddit Dataset. In: Proceedings of the International AAAI Conference on Web and Social Media, Volume 14. 2020; pp. 830–839.
39. Bengio Y, Ducharme R, Vincent P. A Neural Probabilistic Language Model. In: Advances in Neural Information Processing Systems. (Leen T, Dietterich T, Tresp V eds.) 13. 2000.
40. Cer D, Yang Y, Kong S-Y, et al. Universal sentence encoder. arXiv preprint arXiv:1803.11175, 2018.
41. Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and Their Compositionality. In Advances in Neural Information Processing Systems. (Burges CJ, Bottou L, Welling M, et al. eds.) 2013; pp. 3111–3119.
42. Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.
43. Rathore M. Comparison of fasttext and word2vec. Available from: https://markroxor.github.io/gensim/static/notebooks/Word2Vec_FastText_Comparison.html [Last accessed: June 30, 2023].
44. Bennett M. The Financial Industry Business Ontology: Best practice for big data. J Bank Regul 2013;14(3–4):255–268.
45. Roman D, Alexiev V, Paniagua J, et al. The euBusinessGraph ontology: A lightweight ontology for harmonizing basic company information. 2022;13(1):41–68; doi: 10.3233/SW-210424

Cite this article as: Papa M, Chatzigiannakis I, Anagnostopoulos A (2023) Automated natural language processing-based supplier discovery for financial services. *Big Data* 3:X, xxx–xxx, DOI: 10.1089/big.2022.0215.

Abbreviations Used

API = Application Programming Interface
CBOW = Continuous Bag-of-Words
FN = false negatives
FP = false positives
KG = knowledge graph
NED = named-entity–disambiguation
NLP = natural language processing
POS = part-of-speech
TBFY = TheyBuyForYou
TN = true negatives
TP = true positives
USE = Universal Sentence Encoder
w2v = Word2Vec