World Patent Information 73 (2023) 102198

Contents lists available at ScienceDirect

World Patent Information

journal homepage: www.elsevier.com/locate/worpatin

From scattered sources to comprehensive technology landscape : A recommendation-based retrieval approach

Chi Thang Duong [a], Dimitri Perica David [b,c,d], Ljiljana Dolamic [c], Alain Mermoud [c,*], Vincent Lenders [c], Karl Aberer [a]

[a] Distributed Information Systems Laboratory, EPFL, Lausanne, Switzerland
[b] Information Science Institute, University of Geneva, Geneva, Switzerland
[c] Cyber-Defence Campus, armasuisse, Switzerland
[d] Institute of Entrepreneurship & Management, University of Applied Sciences (HES-SO Valais-Wallis), Sierre, Switzerland

ARTICLE INFO

Keywords:
Technology monitoring
Information retrieval
Entity-based retrieval
Technology classifier
Recommender system

ABSTRACT

Mapping the technology landscape is crucial for market actors to take informed investment decisions. However, given the large amount of data on the Web and its subsequent information overload, manually retrieving information is a seemingly ineffective and incomplete approach. In this work, we propose an end-to-end recommendation based retrieval approach to support automatic retrieval of technologies and their associated companies from raw Web data. This is a two-task setup involving (i) technology classification of entities extracted from company corpus, and (ii) technology and company retrieval based on classified technologies. Our proposed framework approaches the first task by leveraging DistilBERT which is a state-of-the-art language model. For the retrieval task, we introduce a recommendation-based retrieval technique to simultaneously support retrieving related companies, technologies related to a specific company and companies relevant to a technology. To evaluate these tasks, we also construct a data set that includes company documents and entities extracted from these documents together with company categories and technology labels. Experiments show that our approach is able to return 4 times more relevant companies while outperforming traditional retrieval baseline in retrieving technologies.

1. Introduction

The expanding and accelerating pace of technology development continuously reshapes the technological landscape [1]. Depicting an up-to-date and holistic map of organizations that develop, implement or sell a given technology is an important business challenge, should a technology be novel or established [2]. In such a dynamic environment, market actors are increasingly confronted by an information overload issue, as an ever raising amount of heterogeneous and unstructured market information needs to be collected, stored, cleaned, structured and analyzed [3]. Hence, the automated analysis of the complex network of organizations and technologies is a key business intelligence necessity not only for public entities, but also for private investors [4].

Such an information retrieval necessity has triggered various business intelligence and technology monitoring procedures, which have been either developed by in-house R&D efforts of organizations or by academic actors (*e.g.*, [5]). Often, such procedures consist of non-automated approaches that struggle to tackle the information overload challenge, as they do not provide a reliable, systematic and scalable information retrieval methodology for mapping the technology market and determine which companies are developing/commercializing which technology [6]. These extant procedures are mainly based on frameworks that find documents matching query terms instead of finding entities *per se* [7]. Yet, finding entities is central when it comes to investigate the technological landscape and finding new technologies. Even though related works have partially investigated such an issue, to the best of our knowledge, no work has provided an end-to-end automated information retrieval framework for finding technology related entities and mapping the technological landscape through a recommendation based perspective. In this work, we develop such a framework in order to map the industrial technology landscape, which would be highly applicable in several domains where a comprehensive understanding of the landscape is required. For instance, in

* Corresponding author.
E-mail address: alain.mermoud@armasuisse.ch (A. Mermoud).

https://doi.org/10.1016/j.wpi.2023.102198
Received 17 January 2022; Received in revised form 9 February 2023; Accepted 18 April 2023
Available online 11 May 2023

cybersecurity, by mapping the technological landscape of cybersecurity, decision-makers will be provided relevant information for taking more informed purchase decisions [8]. Similarly, in the stock market, having a comparative analysis on technological advances of competing companies is crucial for making correct investment decisions.

In this paper, we focus on three important retrieval tasks that are related to technology retrieval. The first task is company to technology retrieval (denoted as com-tech). In this task, given a company name, we would like to return a list of technologies that are related to the company. A technology is related to a company if it is explicitly mentioned in the texts of the company either its website, its tweets or its job postings... A technology can also be implicitly implied by a company by looking at similar companies. Companies that are similar tend to work or leverage a same set of technologies for not losing out in the business competition. The second task is the reverse of the first task where given a technology, we would like to retrieve companies that are related to this technology. This task is denoted as tech-com. The final task involves finding similar companies. Given a company, we would like to find similar companies in terms of their technology portfolio and technology offering. The three tasks are dependent as finding technologies related to a company requires knowing the similarity between companies and vice versa. This motivates us to follow the recommendation-based approach.

A typical example of a recommender system is for recommending movies to users based on past interactions of the users. We can recommend a movie to a user based on the interests of like-minded users or similarity of the movie to the user's watchlist. As a result, there is a strong similarity between the user-movie scenario and our company-technology scenario. Moreover, a non-interaction between a user and a movie does not mean the user does not like the movie but it could just mean that the data is missing or just unobserved. This is similar to the implicitly-observed case of our company-technology retrieval problem. Our proposed approach based on recommendation has the advantage that it enables retrieving, for example, a company that is related to a technology even there is no document that mentions them together. This is possible due to considering the similarities between technologies and between companies. We identify where two companies are similar if they have similar technology portfolio and vice versa, two technologies are similar if they are developed by similar companies. This allows us to say that for instance, a company that is working on machine learning is highly likely to be also working on deep learning.

General framework. Our framework is a two step approach that first classifies the entities into technologies before performing company and technology retrieval. We use DistilBERT [9], which is a state-of-the-art language model, to construct the entity embeddings allowing us to achieve high accuracy in technology classification. For retrieval, we propose a recommendation based approach that takes into account both the technologies, companies and their relationship. This recommendation approach enables better retrieval results in comparison with state-of-the-art learning based recommendation approaches such as GMF [10,11], MLP [10] and NCF [10]. Our results show that our approach is able to return 4 times more relevant companies in company to company retrieval in comparison with traditional tf-idf retrieval approach.

The remainder of this article proceeds as follows. Section 2 grounds this research by emphasizing different methodologies of information retrieval for technology monitoring, and the research gaps. Section 3 details the data collection. Section 4 presents the information retrieval model and framework. Section 5 presents the entity extraction methodology and the technology classification we use. presents the technology retrieval method we used. Section 7 presents the empirical evaluation, while Section 8 concludes and set the path for further research.

2. Related works

The information overload triggered by the big data era has motivated researchers and practitioners to develop numerous automated information retrieval methods by using different yet often complementary approaches [12–14]. Such methods have been widely used in fields as digital libraries [15], information filtering and recommender systems [16,17], media search [18] and search engines [19].

In the field of technology monitoring and forecasting – and more specifically in the specific context of technology landscape monitoring –, numerous works have been published, involving the extraction of patents [20], scientific articles [21] and social media analysis [22]. These technology retrieval methods can be classified into either a *keyword-based* or a *entity-based* approach. Most of the existing methods are keyword-based in which the queries and the data are mostly plain texts [23].

Hossari et al. (2019) proposed an automated framework for detecting the existence of new technologies in texts, and extract terms used to describe new technologies [24]. Aharonson & Schilling developed a framework that captures the distance between patents, and a company's technological footprint. Their framework also enables to measure the proximity of technological footprints between organizations [25]. Tang and Liu (2008) presented a three-layer model for technology forecasting based on text mining techniques, incorporating a collection layer, an analysis layer and a representation layer, before overlapping a semantic web based approach in order to map the industrial technology landscape [26]. On the other hand, entity-based techniques [27] require linking a piece of text to an entity in a knowledge base such that retrieval is done on the entities instead of the raw texts. Woon and Madnick (2008) developed an information retrieval framework for visualizing the technology landscape by exploring the use of term co-occurence frequencies as an indicator of semantic closeness between pairs of entities [28]. In the field of energy related technologies, Mikheev (2018) developed an ontology based data access framework under a semantic approach to query complex datasets, creating an automated mapping procedure to connect data to ontology entities [29]. By applying a semantic approach, Sitarz et al. (2012) developed an automated framework for identifying thematic groups of scientific publications based on clustering of sets of co-occurrence words and financial-analysis techniques for trends detection and forecasting [30].

In our setting, we opt for the entity-based approach as it allows us to handle different mentions of the same entity while enabling better retrieval accuracy due to external information from the knowledge base. However, to the best of our knowledge, little has been done when it comes to apply information retrieval methodologies with the aim of presenting a holistic and comprehensive monitoring and mapping of the industrial technology landscape. In order to do so, a technique needs to consider both the technologies and companies at the same time. The following steps have to be undertaken: (i) an entity fishing approach needs to be applied for extracting and classifying technology entities; (ii) then, these technology entities need to be linked to specific companies; (iii) and finally, these entities must be ranked according to their level of relevance. For instance, in the com-tech retrieval task, the relevance is between the technologies and a specific company while for the com-com retrieval task, the relevance is between the companies. Demartini et al. (2009) provided a formal model for entity fishing and ranking [7]. Yet, to the best of our knowledge, no such work has been deployed in the context of technology landscape monitoring. Similarly, Balog et al. (2012) developed a framework for assessing the strength of association between a topic and a person (*i.e.*, expertise retrieval) [31]. Yet, to the best of our knowledge, no such framework has been applied in the context of technology landscape monitoring for linking technology entities and company entities.

Table 1
Dataset statistics.

Dataset	#comps.	#terms
Website	7907	22,104
Patent	6814	7796
Jobs	3894	2532
Twitter	790	3236
Total	18,339	27,977

While there are a lot of research on leveraging BERT-based model [32] for natural language processing, there are only few papers that focus on the topic of entity extraction in which technology retrieval can be considered as a specific domain. This requires first identifying text spans that are potential candidates for entities and then classify the entities based on their types [33–36]. In [33], the authors propose to capture the relationships between entities using BERT embeddings. This allows them to enumerate and score the entities considering both the local and global contexts. In [34], a holistic approach to extract the entities and their relationships is proposed. It involves identifying the entities in a text using BERT embeddings of text spans. The relationship between entities by classifying the types of the entities and using them to classify the relationships between entities. In [35], the authors propose an end-to-end approach to extract entities and relations using pre-trained language models such as BERT. The entity extraction model is based on the embeddings constructed by BERT for each words in a sentence before applying an entity classifier.

3. Dataset construction

Before discussing our framework, we would like to discuss our process of constructing the dataset. This is important as firstly, to the best of our knowledge, there is no public dataset available for the technology retrieval and classification task. Secondly, since we are publishing the dataset, describing the data collection process clearly would be helpful for potential users of our dataset. While there are several datasets that are in the domain of patent retrieval tasks such as CLEF-IP [37], NTCIR [38], they are not applicable in our setting as our tasks (com-tech, com-com, tech-com) are not specific to patents but more about technologies. Our tasks require having a dataset that contain information about the companies and the technologies.

As a result, we aim to create and publish such a dataset to further research in this field. We have defined the following requirements for our dataset. First, the dataset should be multilingual as the technology retrieval task should be language independent. Second, the dataset should be realistic and coming from real-world data as this would enable objective evaluation of any proposed approach.

In the following, we discuss our data construction process. The dataset is constructed by first crawling different data sources that are publicly available on the Internet. As we aim to develop a language-agnostic framework, we decide to construct a dataset based on Swiss companies as Switzerland is a multilingual country with French, German and Italian as official languages while English is a working language. For each company, we intend to collect all possible documents that are related to the company's actual activities. This involves the following data sources: the company's *website*, the company's *job postings*, the *patents* and the company's *tweets*. In addition, to maintain an up-to-date dataset about the companies, we periodically crawl data from the above data sources. The above data sources are selected as they could provide different perspectives regarding a company's technology offering. The statistics of the dataset[1] is shown in Table 1. In this table, the total numbers represents the number of unique companies and terms.

[1] Note that there are overlapping companies and terms among different data sources.

3.1. Collecting data

To collect the data, we first need a list of Swiss companies. We obtain the list of companies registered in Switzerland from the federal Central Business Name Index (Zefix).[2] Additional information regarding given company is then extracted from the corresponding cantonal commercial register record. These records provide, among others, information on location, people, type of company.

Websites. As commercial registers do not provide the information on regarding websites of the registered companies, we need to find the company's website based on the company's name. We first clean the company name (e.g. removing GmbH, Sàrl, Sagl, AG ...) before performing Google search with the remaining terms. Title and description of the first 10 results of this search are extracted. In addition the we crawl the first page of the top level domain (e.g. "http://www.domain.ch" for the search result "http://comp.domain.ch/about") for each of the search results to extract additional information. The information extracted from commercial registers in combination with the results of the previously mentioned search are then put through the classifier to link the company to a correct website. We then crawl the pages from the detected website, using their text for entity extraction.

Patents. We use Patents data from United States Patent and Trademarks Office (USPTO). We use USPTO instead of Patstat or WIPO since it provides refined patent dataset. PatentView,[3] project sponsored by the USPTO to clean and refine the patent database and make it available for research purposes, provides half yearly data dump of the USPTO database, which we inject directly into our system. The patents are linked to the companies based on the location and assignee information, while the title and the abstract of the patents are used to extract the entities.

Jobs. The Indeed[4] is used as a source of information related to jobs. We perform weekly per Swiss canton search for jobs, retrieving for each job the following information: title, description, company name and the original posting. Job's title and description are used for technology annotation, while the linking to the company is based on it's name and location. Indeed removes automatically the duplicate jobs postings and allows for the re-posting of the same job with 60 days delay. We rely on these features for the data curation, since we use this platform as the unique data source for jobs.

Tweets. For each company, we look up its Twitter handle from Crunchbase which is a commercial database of company information. From the handle, we collect the latest 3000 tweets for each company using sempi.tech which is a social listening framework. We use these tweets for entity extraction.

3.2. Entity extraction

From the documents collected in the previous step, we use DBPedia Spotlight (DBPS) [39], an open source tool to automatically annotate the mentions of the DBPedia entities within the text content. DBPedia is a multi-domain ontology derived from Wikipedia. Each DBPedia entity is an URI with the prefix http://dbpedia.org/resource/ followed by the identifier of the corresponding Wikipedia article. DBPS allows us to not only extract entities but also link the entities to their corresponding DBPedia entities.

DBPS is also capable of handling multiple languages, which is important in our setting. In more detail, DBPS can identify entities in different languages and it can map each of them to a DBPedia entry. As DBPedia is a knowledge graph, DBPedia entries are connected. For instance, DBPedia entries of the same entity in different languages are connected through the relationship predicate *owl:sameAs*. Given a

[2] https://zefix.ch
[3] https://patentsview.org
[4] http://indeed.ch

DBPedia entry of the entity "Pomme" in French,[5] we can trace back to its English equivalence[6] through the sameAs connection. This allows us to handle similar terms in different languages.

For each data source, we perform entity extraction independently. The output of this step is a list of entities and their corresponding number of appearance in the data source of a company. We store the output in JSON line format where each line is a triple of company, DBPedia entity and number of entity occurrences. Note that while our data sources are heterogeneous as they come from different domains, DBPS allows us to standardize them. Whether it is a website, a patent, a job posting or a tweet, DBPS extracts all the candidate technology mentions while ignoring irrelevant data. These technology mentions are the input to any technology classification or retrieval technique.

3.3. Data labeling

Technology labeling. To support the evaluation of a technology classifier, we also provide a list of Wikipedia terms and their labels whether a term is about a technology or not. The articles of Wikipedia are classified by Wikipedia to belong to different Main Topic Classifications [40] such as Technology, Science or Engineering. Each MTC is further divided into subcategories and each subcategories can also be divided further. At the leaf level of these categorization trees are the Wikipedia articles. Even though Technology is one of Wikipedia's MTC categories, one cannot rely on this concept to extract all technology related articles, as the categories within the Wikipedia graph are very loosely related ("is related to"). We approach the labeling in a top-down approach where we aim to label the MTC categories. First, Wikipedia directed categories graph was cleaned by removing hidden categories, admin and user pages, followed by regular expression filters removing categories referring to companies, brands, currencies etc. We than calculate the shortest path to each of the 28 MTC, and retain the categories having the shortest path to Technology, Science, or Engineering topics. This process resulted in the list of 7876 categories. These categories were than manually labeled as technology or non technology, resulting with 1356 categories being labeled an technology. This is the only manual step in our data construction process. An article is then considered to be a technology if it is directly connected to a category labeled as such. In other words, an article is directly connected to a category if there is a path from the article that goes up to the category in the categorization tree.

Company categorization. Crunchbase maintains a database of companies and their detail information where the data are manually curated by Crunchbase staff and online contributors.[7] Each company is associated with several categories describing its main activities. For instance, Roche which is a pharmaceutical company based in Switzerland is categorized as Biotechnology, Health Care, Health Diagnostics and Pharmaceutical. From the companies collected in the above steps, we crawl Crunchbase to obtain their categories. The categories can be considered as pseudo-labels for our com-com and tech-com retrieval tasks.

4. Model and approach

We develop a unified framework to classify entities into technologies and to perform technology related retrieval. This requires solving two tasks of technology classification and technology retrieval.

4.1. Model

Our framework considers a set of companies $C = \{c_1, \ldots, c_n\}$ and a set of entities $E = \{e_1, \ldots, e_m\}$. The connection between an entity e_i and a company c_j can be observed through several data sources. For each data source, we measure this connection by the number of times the entity e_i is mentioned by the company c_j. We can present these connections for a specific data source by an *interaction matrix* $\boldsymbol{M}$ where M_{ij} is the occurrence frequency of the entity e_i in the corpus of company c_j. We also denote these matrices by $M = \{\boldsymbol{M}_1, \ldots, \boldsymbol{M}_k\}$ where $\boldsymbol{M}_i$ represents an interaction matrix where k is the number of data sources. Each interaction matrix captures the interactions between companies and entities in a data source.

Tasks. We have 3 retrieval tasks: company to technology (com-tech) retrieval, company to company (com-com) retrieval and technology to company (tech-com) retrieval. Com-tech and com-com retrieval are connected as both require an accurate representation of a company by its technologies. The representation of a company by its technologies in its simplest could be the set of technologies or its sparse vector equivalence. A more sophisticated representation would involve representing each company by a dense vector i.e. an embedding. Using an embedding to represent each company and each technology is the approach we use in this paper. Two companies are similar if they have similar technology representation. Similarly, for tech-com retrieval, we need a good representation of a technology by its companies. There is a mutual reinforcing relationship between company and technology representation. A good company representation requires knowing similar technologies while knowing company similarity is helpful in constructing a good technology representation. This means we need a common model to approach these three different retrieval tasks.

4.2. General approach

Our framework takes as input the entities extracted from the company corpora. These entities are identified using entity extraction frameworks such as DBPedia-spotlight. Each entity is associated with its description. This information is used in the second step to classify the entities into technologies or not. Our technology classifiers are constructed using BERT as a feature extractor. The technologies obtained from the previous step are used as input for the retrieval step. We reformulate the retrieval problem as a recommendation problem based on collaborative filtering where the technologies are "recommended" to a company if the technologies are considered to be related to the company's activities. Casting this as a recommendation problem has several benefits. The relevancy of a technology to a company can be measured more accurately if similar companies in the same domain are considered. This also means that missing technologies in a company corpus can be recovered by considering similar companies. We propose a technology recommendation model that extends traditional matrix factorization by integrating both the semantics/meaning of a technology and the interactions between technologies and companies. Our BERT-based model for technology classifier can identify if two technologies are closely related such as deep learning and machine learning while our recommendation-based model can capture the interactions between technologies and companies e.g. whether a company mentions a technology (see Fig. 1).

5. Technology classification

As the input entities to our system are extracted using an entity extraction framework such as DBPedia-spotlight [41], they cover all possible domains while we are interested only in technologies. To this end, we develop a technology classifier to filter out unrelated entities. Note that DBPedia-spotlight also performs entity linking where each entity is linked to a DBPedia page or a Wikipedia article describing this entity. We leverage this description to construct our technology

[5] https://fr.dbpedia.org/page/Pomme
[6] https://dbpedia.org/page/Apple
[7] crunchbase.com

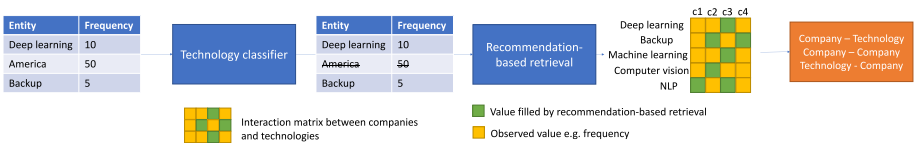

Fig. 1. From entities to company and technology retrieval.

classifier. For each entity, we extract its abstract which we define to be the description from DBPedia or the first paragraph of its Wikipedia article whichever available. Note that in this work, we aim to focus on classifying the technologies based on the textual contents of the corpus containing them in a uniform way in an end-to-end manner. Additional information are available such as the patent classification of the Patents dataset could be used to augment the technology classification step. However, we leave this as a future work as combining an existing classifier with an entity-based classifier requires careful mapping between them.

BERT-based encoder. We use DistilBERT [9] which is a state-of-the-art language model while being light-weight and fast to train to construct abstract embeddings. We use these abstract embeddings as representations for the entities. We pass each abstract through DistilBERT to obtain the token embedding of the [CLS] token, which is commonly used as the sentence or paragraph embedding. We denote this token embedding as $\boldsymbol{z}_e$ where e denotes an entity.

While there are other word embedding models such as Glove or word2vec, we employ BERT-based models such as BERT, DistilBERT as the word embeddings from these models are context-dependent. This is extremely helpful in disambiguating words that have many meanings. For instance, the word "cookies" as a technological term and "cookies" as a biscuit would have the same word embedding in traditional model such as Glove or word2vec. However, in contextual language models such as DistilBERT, the word "cookies" would have different embeddings depending on its context i.e. surrounding words in a sentence. In other words, contextual word embeddings from BERT-based models capture more semantics which is helpful in detecting technologies.

Although DistilBERT is a distilled version of BERT which was trained on English Wikipedia, distillation may incur information lost. To this end, we propose to fine-tune DistilBERT to better capture the abstract meaning which usually contain several scientific terminology, therein, obtaining better entity representation. We fine-tune DistilBERT for our technology classification in an end-to-end manner where we feed the abstracts through the model to obtain the abstract embeddings. These abstract embeddings are then fed through a linear classifier to get the technology predictions. The parameters of DistilBERT and the linear classifier are updated together in an end-to-end manner using SGD [42] with the binary cross entropy loss as the loss function. We observe that with finetuning, we are able to achieve better accuracy.

Embedding refinement. To further increase the capacity of our classifier, we pass $\boldsymbol{z}_e$ through several layers including a dropout layer to reduce overfitting. More precisely, $\boldsymbol{z}_e$ is passed through a multi-layer neural network with 2 blocks where each block is a linear layer followed by a BatchNorm layer and a sigmoid non-linearity. A block can be formulated as follows:

$$\boldsymbol{t}_e = \sigma(BatchNorm(\boldsymbol{W}\boldsymbol{z}_e + b_1))$$

Between 2 blocks, we also use a dropout layer to reduce overfitting. We observe that be refining the embedding this way, we can achieve better accuracy than non-refinement.

6. Technology retrieval

The case for retrieving as recommending. There are several requirements for the retrieval model of this task. First, the technologies retrieved for a company must be derived from the technologies mentioned in the company's corpus as these technologies are the most likely ones to reflect the company's actual activities. As each technology has a different level of relevancy to the company, com-tech retrieval considering only observed technologies is a *technology ranking* problem. Second, it is safe to assume that a company's corpus may not contain all the technologies the company is working on. Companies may not publicly mention a technology to keep a competitive advantage or it could simply be due missing data. For these reasons, the com-tech retrieval is also a *technology discovery* problem. To solve both problems at the same time, we propose a recommendation model that identifies potentially related technologies of a company by looking at similar companies while measuring the relevancy between every pair of technology and company for ranking technologies.

Recommendation model. Given a set of companies $\mathbb{C} = \{c_1, \dots, c_n\}$ and the set of technologies $\mathbb{T} = \{t_1, \dots, t_m\}$ obtained after the technology classification step, our recommendation model takes as input a com-tech interaction matrix $\boldsymbol{M} \in \mathbb{R}^{n \times m}$ as

$$M_{ij} = \begin{cases} f(c_i, t_j), \text{if there is a mention of } t_j \text{ in any data source of } \mathbb{C}_i \\ \emptyset, \text{if there is no mention of } t_j \text{ in } \mathbb{C}_i \end{cases}$$

The function $f(c_i, t_j)$ captures the importance of t_j according to c_i. This importance can be measured by the number of times t_j occurs in the corpus of c_i rescaled by some weighting scheme. In our setting, we measure the importance per data source using tf-idf before combining all data sources:

$$f(c_i, t_j) = \sum_k w_k f_k(c_i, t_j)$$

where $f_k(c_i, t_j)$ is the importance function of data source k which is measured by the tf-idf value of t_j considering each company as a "document". w_k is its associated weight which captures our perceived relevance of the data source to our retrieval tasks.

A high value of $f(c_i, t_j)$ does not mean the company c_i is actually working on technology t_j. For instance, during the pandemic, there could be several mentions of the word "vaccine" but it does not necessarily mean that a certain company is developing a vaccine. This example shows that answering com-tech retrieval by only looking at the company corpus could be problematic. We can have the same argument for $M_{ij} = \emptyset$. This does not mean the company c_i has no activity related to t_j. It could be the case that the company is working on this technology but it has not mentioned it *yet* in the corpus.

The above com-tech interaction model is akin to collaborative filtering with implicit feedback [10,11]. To answer the com-tech retrieval problem, we first need to solve the recommendation problem where we need to estimate the unobserved entries of $\boldsymbol{M}$. The estimation is usually done by learning a model $\hat{M}_{i,j} = f(c_i, t_j|\Theta)$ where $\hat{M}_{i,j}$ is the estimated score of M_{ij}, f is a parameterized function that predicts the interaction score between c_i and t_j, Θ denotes the parameters of f.

Semantic-aware matrix factorization. We propose a semantic-aware recommendation model extending traditional matrix factorization (MF) approach. In MF, each company and each technology is represented by an embedding in a shared latent space. The technology and the company embeddings are learned such that they can reconstruct the interaction matrix $\boldsymbol{M}$. More precisely, let $\boldsymbol{c}_i \in \mathbb{R}^d$ and $\boldsymbol{e}_j \in \mathbb{R}^d$ be the embeddings of company c_i and technology t_j respectively. Then, MF aims to estimate the relevancy score M_{ij} by:

$$\hat{M}_{ij} = f(c_i, t_j|\boldsymbol{c}_i, \boldsymbol{e}_j) = \boldsymbol{c}_i^T \boldsymbol{e}_j$$

However, MF considers the technologies to be independent even if they are semantically related such as "deep learning" and "machine learning". To this end, we propose to incorporate the meaning of the technologies into MF while extending the model capacity by passing the technology embedding through several linear layers. In the following, we describe in detail the layers of our architecture.

Semantic embedding: We capture the technology meaning using BERT [9,32] as a feature extractor over the technology abstract:

$$\boldsymbol{s}_i^{(0)} = f_{BERT}(a_i)$$

where $\boldsymbol{s}_i$ is the semantic technology embedding and a_i is the abstract of technology t_i.

MLP layers: The semantic technology embedding is passed through several MLP layers to further reduce the size of the embedding while increasing the model capacity.

$$\begin{aligned} \boldsymbol{s}_i^{(1)} &= \boldsymbol{W}_1 \boldsymbol{s}_i^{(0)} + b_1 \\ &\cdots \\ \boldsymbol{s}_i^{(k)} &= \boldsymbol{W}_k \boldsymbol{s}_i^{(k-1)} + b_k \end{aligned} \tag{1}$$

where $\boldsymbol{W}_i, b_i, \sigma$ are the weight matrix, the bias and the non-linearity.

Combination layer: To obtain the final technology embedding $\boldsymbol{t}_i$, we combine the semantic technology embedding $\boldsymbol{s}^{(k)}$ with the raw technology embedding from MF $\boldsymbol{e}_i$ by summing them: $\boldsymbol{t}_i = \boldsymbol{s}^{(k)} + \boldsymbol{e}_i$. The summation allows us to save model's parameters in comparison with concatenation while it is also inspired by transformer architecture [32, 43] where positional encodings are added to the word embeddings.

Model learning. To learn the parameters, traditional MF uses the squared loss between the predicted and actual interaction score: $\mathcal{L}_{sqr}(\Theta) = \sum_{c_i,t_j \in \boldsymbol{M}} (M_{ij} - \hat{M}_{ij})$. However, such a method does not consider the unobserved entries directly. To this end, we follow the pairwise learning approach that aims to optimize the relative ranking between technologies. We use the margin hinge loss which is defined as follows:

$$\mathcal{L}_{hinge}(\Theta, c_i, t_j, t_k) = max(0, m + M_{ij} - \hat{M}_{ik})$$

where c_i, t_j is an observed pair of company and technology while t_k is a negative sample meaning c_i, t_k is an unobserved entry of the interaction matrix $\boldsymbol{M}$.

Recommendation-based Retrieval. Then, the com-tech retrieval problem can be answered by ranking the technologies $\mathbb{T}$ with respect to a company c based on their interaction scores. More precisely, let $\boldsymbol{N}$ to be the com-tech interaction matrix after the unobserved entries are estimated. The top-k com-tech retrieval result for a company c_i is a list of technologies ordered by their interaction scores $\boldsymbol{N}$.

7. Empirical evaluation

7.1. Experimental setup

Datasets. We evaluate our model on the constructed dataset described in Section 3. As our system crawls new data from all data sources constantly which makes it difficult for evaluation, we fix the dataset used in the experiments to be the data collected before 01/04/2020. This snapshot and the code are publicly available at https://figshare.com/s/c014bb8565705e74dd1b.

Metrics. To evaluate the results of com-com retrieval, we leverage the company categorization from Crunchbase. We measure the quality of com-com retrieval by the number of overlapping categories between the query company and the results. More precisely, let $\mathbb{C}(c)$ denote the set of categories of company c. We define the retrieval accuracy for a company c (i.e. the number of overlapping categories) considering top-k most relevant results as follows:

$$P@k(c) = \frac{\sum_{i=\overline{1,k}} |\mathbb{C}(c) \cup \mathbb{C}(c_i)|}{k} \tag{2}$$

Table 2
Comparison of technology classifiers.

	F1-score	Accuracy	AUC
tf-idf	0.531	0.781	0.819
DistilBERT w/o refine w/o finetune	0.499	0.771	0.811
DistilBERT + refine + w/o finetune	0.597	0.734	0.806
DistilBERT + refine + w/ finetune (Ours)	**0.639**	**0.799**	**0.857**

This metric can be extended to a set of companies C as $P@k(C) = \frac{\sum_{c \in C} P@k(c)}{|C|}$.

While it is "straightforward" to evaluate the search results for com-com, the evaluation for com-tech and tech-com is more challenging as there is no available ground truth. For tech-com search, we follow the approach of com-com retrieval where we label each technology by the Crunchbase categories. This is akin to consider each technology as a "company". The number of technologies to be labeled is usually small as we are interested in only important technologies. To this end, we have labeled 119 technologies which are considered important in the cybersecurity domain [44]. The retrieval accuracy $P@k(t)$ for a technology t is defined similarly as in Eq. (2). On the other hand, for com-tech retrieval, this approach is not practical as for each company, the list of retrieved technologies is very large. To this end, we opt for a qualitative evaluation.

Baselines. For technology classification, we construct a baseline using SVM on tf-idf featurization. More precisely, we construct a vector representing a Wikipedia category by combining the vector distances of a category to each of the Wikipedia's MTC and its TF-IDF weighted bag of words (BOW) representation. The weighted BOW representation of the given category is created from the stemmed text obtained by concatenating the abstracts of all Wikipedia articles directly connected to it. Mutual information based feature reduction than resulted in a vector of the length 1000. These vectors are used as the input features for the classifier. For retrieval, we first compare with a tf-idf retrieval approach where the tf-idf values are also the relevancy of the technologies in com-tech retrieval. For com-com retrieval, it is an tf-idf weighted version of Jaccard similarity where each company is represented by its set of technologies. We also compare our recommendation-based retrieval approach with other recommendation models including GMF [10,11], MLP [10] and NCF [10]. The above baselines are selected as to the best of our knowledge, there is no public implementation of technology classification and retrieval techniques. The above baselines represent the best starting points for these tasks.

Environments. Our experiments ran on an Intel Xeon CPU E5-2620 v4 @ 2.10 GHz server with a Titan V GPU with 12 GB VRAM and 128 GB RAM. Our model was implemented using Pytorch 1.7.1 and Spotlight as the recommendation framework and DistilBERT from HuggingFace as the language model.

7.2. Effectiveness of technology classification

Quality of technology classifiers. In this experiment, we analyze the correctness of our technology classifiers. We compare our proposed classifier using BERT with the baseline classifiers based on tf-idf and other BERT models. For this experiment, we compare these approaches on three metrics: accuracy, f1-score and AUC. We use k-fold cross validation with a 80–20 split. We compare our approach which includes using DistilBERT with embedding refinement and finetuning (denoted by DistilBERT + refine + w/ finetune). For the baselines, we perform ablation study to analyze the effectiveness of using embedding refinement and finetuning. These baselines are denoted by DistilBERT + refine + w/o finetune and DistilBERT w/o refine w/o finetune. The last baseline is tf-idf as discussed in Section 7.1.

The results show in Table 2 confirms the benefit of fine-tuning and our refinement step. Our proposed approach outperforms the baselines on all metrics. The difference between using tf-idf as feature and BERT

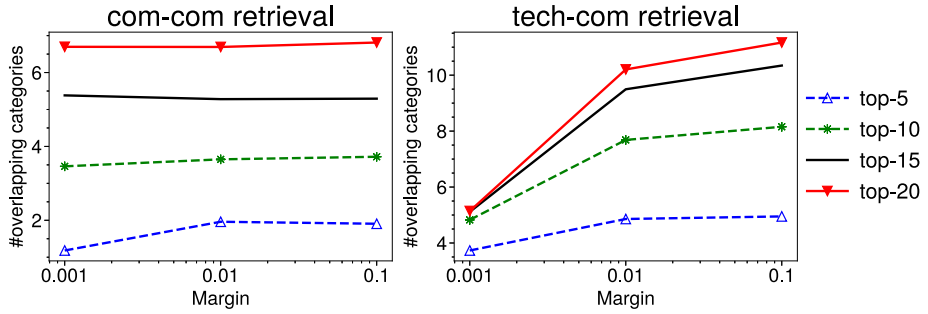

Fig. 2. Effects of margin.

Table 3
Effects of retrieval model.

	Company-company retrieval				Technology-company retrieval			
	top-5	top-10	top-15	top-20	top-5	top-10	top-15	top-20
MF	3.121	3.556	3.568	3.568	1.078	1.882	2.816	3.461
MLP	3.215	3.819	3.836	3.836	1.065	2.053	3.079	3.539
NCF	2.905	3.103	3.105	3.105	1.118	2.211	2.947	3.5
BERT	4.155	5.986	6.636	6.722	1.986	3.211	3.882	5.066
MF+BERT	**4.458**	**7.004**	**8.447**	**8.845**	**2.197**	**3.684**	**5.105**	**6.289**

Table 4
End-to-end evaluation.

	Com-com retrieval		Tech-com retrieval	
	tf-idf	Ours	tf-idf	Ours
top-5	1.2916	**4.458**	2.0161	**2.197**
top-10	1.3022	**7.004**	3.4838	**3.648**
top-15	1.3022	**8.447**	4.3710	**5.105**
top-20	1.3022	**8.845**	4.7419	**6.289**

is 0.1, 0.02 and 0.04 for F1-score, accuracy and AUC respectively. This is expected as large language models trained on large text corpora are able to capture word meanings better. We also observe that fine-tuning improves accuracy in comparison with using BERT without fine-tuning.

7.3. Effectiveness of technology retrieval

In this experiment, we analyze our proposed recommendation-based retrieval model. We compare our model with a tf-idf based retrieval where each technology is associated with a tf-idf value while each company is represented by a tf-idf vector. We also compare our approach with several recommendation models including (1) Generalized MF [11] which is a generalized version of MF, (2) MLP [10] which is a multi-layer recommendation model starting from random vectors and (3) NCF [10] or Neural Collaborative Filtering which is a recommendation model based on deep learning.

The experimental results shown in Table 3 show that our proposed model based on BERT embeddings as initial technology embeddings are better than the baselines. The difference between our worst model and the best baseline is 0.6 at top-5 for com-com retrieval and 0.8 at top-5 for tech-com retrieval. This can be explained by the fact that our models can capture the meaning of the technologies while the baselines consider the technologies to be independent. Among our proposed models, adding the raw technology embedding from MF with the BERT technology embedding is better than using the BERT embedding alone. We can attribute this to the increase the number of parameters of our model i.e. larger capacity which helps in capture the interaction matrix better. The increased capacity is equal to the number of companies and technologies times its embedding size.

7.4. Parameter sensitivity

Effects of margin. We vary the margin of the hinge loss from 0.001 to 0.1 to analyze its effects on the retrieval results. The experimental results are shown in Fig. 2. We observe that the number of overlapping categories tends to increase with the margin. For instance, the number of overlapping categories for top-5 com-com retrieval is 3.73 when the margin is 0.001 but it increases to 4.95 when the margin is 0.1. This is expected as with the larger margin, our model tends to generalize as it aims to capture common technologies between the companies. We observe this phenomenon clearly from Table 5 that we obtain more specific technologies with smaller margin. With larger margins, generic technologies that are shared among different companies are more representative than more specific ones. This experiment confirms our ability to control the specificity of the retrieval results by changing the margin of the hinge loss.

Embedding size. In this experiment, we analyze the effects of the embedding size on the retrieval results. We vary the embedding size from 32 to 512. Results in Fig. 3 shows that as the embedding size increases, we can retrieve companies better for both tech-com and com-com retrieval tasks. This is expected as increasing embedding size also improves the model capacity. However, there is a trade-off in increasing embedding size as it incurs longer training time as shown in Fig. 4. The difference in training time between embedding size of 32 and 512 is 3 times. However, even with the largest embedding size, the training time per epoch is still very fast — only around 1.5 s.

7.5. End-to-end comparison

Having evaluated the individual components of our solution, we turn to its end-to-end performance in comparison with the baseline. Table 4 compares the performance of our approach with a tf-idf based retrieval approach which uses tf-idf as feature for technology classifier

Table 5
Com-tech retrieval.

Acronis AG			InterHype SARL		
Ours-0.01	Ours-0.1	tf-idf	Ours-0.01	Ours-0.1	tf-idf
CyberTruck	Encryption	Cloud computing	Computer security	Cloud computing	Nous
Virtual machine	Communication	Backup	Automatic train protection	Computer science	Sand
Cloud storage	Virtualization	Disaster recovery	SMS	Computer security	Antiseptic
Off-site data protection	Internet	Web server	Off-site data protection	Digital transformation	Habitat
Encryption	Personal firewall	Ransomware	Backup	Information security	Glass

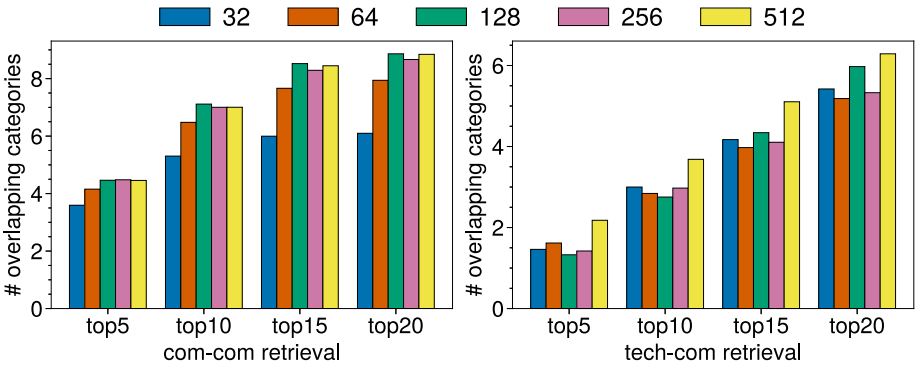

Fig. 3. Effects of embedding size.

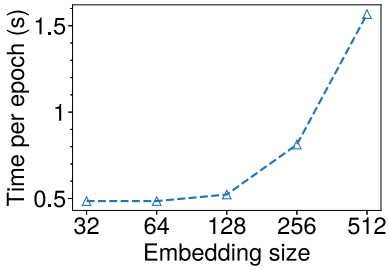

Fig. 4. Training time per epoch.

and technology retrieval. Our approach denoted by Ours represents our end-to-end framework which uses DistilBERT for technology classification and recommendation-based approach for technology retrieval. Our approach leads to significantly better retrieval results in both retrieval tasks. Our model is nearly 4 times better than the baseline in the com-com retrieval at top-5 while the difference is 0.18 for tech-com retrieval. First, this can be attributed to our approach's better technology classifier by using language model. Second, our recommendation retrieval model considers both the companies, technologies and their relationships as the same time, while the technologies and companies in the tf-idf model are handled independently. This enables our model to leverage the similarity of companies to support technology retrieval and vice versa.

7.6. Qualitative analysis

In this experiment, we analyze the retrieval results qualitatively between our approach (denoted by Ours as in Section 7.5) and the tf-idf baseline. Table 5 shows the com-tech retrieval results where we search for cybersecurity companies. For com-tech retrieval, as discussed above, we are able to control the specificity of the results by changing the margin. In addition, our proposed model is able to return less noise in comparison with tf-idf one as tf-idf model may return non-technological terms due to the quality of its classifier. This phenomenon can be seen for instance in the search for InterHype Sarl which is a cybersecurity company. For com-com and tech-com retrieval, due to space constraint, we do not include them. However, we observe that our model can return companies that are in the same domains as the queried company or technology. This is in line with the quantitative result observed in previous experiments.

8. Conclusion and future work

In this paper, we propose an end-to-end framework to first extract and classify technological mentions from company corpuses and then, retrieve related technologies and companies. Our technology classifier is based on DistilBERT model with finetuning and refinement to achieve better accuracy while our recommendation-based retrieval model enables more relevant results. We envision that our framework can also be used for other retrieval tasks where we want to extract terms that belong to a specific domain e.g. business-related terms. This would require obtaining new training data for this domain and retrain the models.

Limitations and Future work. First, as entity extraction is not a part of our framework. The entity extraction step and technology classification are done independently. In doing so, the technology classification step does not have access to the contexts of the entities. This would reduce the accuracy of the classification step. We would like to combine the technology classification and entity extraction step for more accurate results. Second, our current approach does not allow users to reformulate the queries. Queries posed by users in the same session would be considered independent. In general, by allowing query reformulation, we would better capture user's intention. This would lead to better retrieval results.

CRediT authorship contribution statement

Chi Thang Duong: Conceptualization, Methodology, Software, Investigation, Writing. **Dimitri Perica David:** Conceptualization, Writing – original draft, Writing – review & editing. **Ljiljana Dolamic:** Conceptualization, Software, Investigation, Writing – original draft, Writing – review & editing. **Alain Mermoud:** Conceptualization, Writing – review & editing, Project administration. **Vincent Lenders:** Conceptualization, Resources, Supervision, Project administration. **Karl Aberer:** Conceptualization, Resources, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Shalf, The future of computing beyond Moore's Law, Phil. Trans. R. Soc. A 378 (2166) (2020).
[2] P. Rikhardsson, O. Yigitbasioglu, Business intelligence & analytics in management accounting research: Status and future focus, Int. J. Account. Inf. Syst. 29 (2018) 37–58.
[3] P.G. Roetzel, Information overload in the information age: A review of the literature from business administration, business psychology, and related disciplines with a bibliometric approach and framework development, Bus. Res. 12 (2) (2019) 479–522.
[4] U. Durak, Advances in Aeronautical Informatics, Springer, 2018, pp. 3–13.
[5] T.U. Daim, D. Chiavetta, A.L. Porter, O. Saritas, Anticipating Future Innovation Pathways Through Large Data Analysis, Springer, 2016.
[6] D. Loveridge, C. Cagnin, Anticipating Future Innovation Pathways Through Large Data Analysis, Springer, 2016, pp. 3–23.
[7] G. Demartini, J. Gaugaz, W. Nejdl, A vector space model for ranking entities and its application to expert search, in: ECIR, Springer, 2009, pp. 189–201.
[8] T. Mahmood, U. Afzal, Security analytics: Big data analytics for cybersecurity: A review of trends, techniques and tools, in: 2013 2nd National Conference on Information Assurance, Ncia, IEEE, 2013, pp. 129–134.
[9] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter, 2019, arXiv preprint arXiv:1910.01108.
[10] X. He, et al., Neural collaborative filtering, in: WWW, 2017, pp. 173–182.
[11] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, in: UAI, 2009, pp. 452–461.
[12] K. Munir, M.S. Anjum, The use of ontologies for effective knowledge modelling and information retrieval, Appl. Comput. Inform. 14 (2) (2018) 116–126.
[13] P. Kaur, H.S. Pannu, A.K. Malhi, Comparative analysis on cross-modal information retrieval: A review, Comp. Sci. Rev. 39 (2021) 100336.
[14] W. Shalaby, W. Zadrozny, Patent retrieval: A literature review, Knowl. Inf. Syst. (2019) 1–30.
[15] W.R. Hersh, Biomedical Informatics, Springer, 2014, pp. 613–641.
[16] N.J. Belkin, W.B. Croft, Information filtering and information retrieval: Two sides of the same coin? Commun. ACM 35 (12) (1992) 29–38.
[17] I.A. Heggo, N. Abdelbaki, Intelligent Systems in Big Data, Semantic Web and Machine Learning, Springer, 2021, pp. 23–49.
[18] J. Rao, W. Yang, Y. Zhang, F. Ture, J. Lin, Multi-perspective relevance matching with hierarchical convnets for social media search, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, (no. 01) 2019, pp. 232–240.
[19] C.C. Aggarwal, Machine Learning for Text, Springer, 2018, pp. 259–304.
[20] M. Kim, Y. Park, J. Yoon, Generating patent development maps for technology monitoring using semantic patent-topic analysis, Comput. Ind. Eng. 98 (2016) 289–299.
[21] F. Dotsika, A. Watkins, Identifying potentially disruptive trends by means of keyword network analysis, Technol. Forecast. Soc. Change 119 (2017) 114–127.
[22] X. Chen, T. Han, Disruptive technology forecasting based on gartner hype cycle, in: TEMSCON, IEEE, 2019, pp. 1–6.
[23] K. Arai, Extraction of keywords for retrieval from paper documents and drawings based on the method of determining the importance of knowledge by the analytic hierarchy process: AHP, Int. J. Adv. Comput. Sci. Appl. 11 (10) (2020).
[24] M. Hossari, S. Dev, J.D. Kelleher, TEST: A terminology extraction system for technology related terms, in: Proceedings of the 2019 11th International Conference on Computer and Automation Engineering, 2019, pp. 78–81.
[25] B.S. Aharonson, M.A. Schilling, Mapping the technological landscape: Measuring technology distance, technological footprints, and technology evolution, Res. Policy 45 (1) (2016) 81–96.
[26] F. Tang, Y. Liu, Applying semantic web into technology forecasting in enterprises, in: International Conference on Service Operations and Logistics, and Informatics, vol. 1, IEEE, 2008, pp. 135–138.
[27] Y. Wang, M. Rastegar-Mojarad, R. Komandur-Elayavilli, H. Liu, Leveraging word embeddings and medical entity extraction for biomedical dataset retrieval using unstructured texts, Database 2017 (2017).
[28] S. Madnick, W.L. Woon, Semantic distances for technology landscape visualization, 2008.
[29] A. Mikheev, Ontology-based data access for energy technology forecasting, in: IWCI 2018, Atlantis Press, 2018, pp. 147–151.
[30] R. Sitarz, A. Kraslawski, Computer Aided Chemical Engineering, vol. 30, Elsevier, 2012, pp. 437–441.
[31] K. Balog, Y. Fang, M. de Rijke, P. Serdyukov, L. Si, Expertise retrieval 6 (2–3), 2012, pp. 127–256.
[32] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.
[33] D. Wadden, U. Wennberg, Y. Luan, H. Hajishirzi, Entity, relation, and event extraction with contextualized span representations, 2019, arXiv preprint arXiv:1909.03546.
[34] Z. Zhong, D. Chen, A frustratingly easy approach for entity and relation extraction, 2020, arXiv preprint arXiv:2010.12812.
[35] J. Giorgi, et al., End-to-end named entity recognition and relation extraction using pre-trained language models, 2019, arXiv preprint arXiv:1912.13415.
[36] X. Du, A.M. Rush, C. Cardie, GRIT: Generative role-filler transformers for document-level event entity extraction, 2020, arXiv preprint arXiv:2008.09249.
[37] F. Piroi, M. Lupu, A. Hanbury, Overview of CLEF-IP 2013 lab, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2013, pp. 232–249.
[38] T. Yamamoto, Z. Dou, Overview of NTCIR-16, in: Proceedings of the 16th NTCIR Conference on Evaluation of Information Access Technologies, National Institute of Informatics, Tokyo, Japan, 2022.
[39] J. Daiber, M. Jakob, C. Hokamp, P.N. Mendes, Improving efficiency and accuracy in multilingual entity extraction, in: ICSS, New York, NY, USA, ISBN: 9781450319720, 2013, pp. 121–124.
[40] Wikipedia main topic classfication, 2021, https://en.wikipedia.org/wiki/Category:Main_topic_classifications. (Accessed 16 June 2021).
[41] P.N. Mendes, M. Jakob, et al., DBpedia spotlight: Shedding light on the web of documents, in: Proceedings of the 7th International Conference on Semantic Systems, 2011, pp. 1–8.
[42] S. Ruder, An overview of gradient descent optimization algorithms, 2016, arXiv preprint arXiv:1609.04747.
[43] A. Vaswani, et al., Attention is all you need, NeurIPS 30 (2017) 5998–6008.
[44] Security-relevant technology and industry base, 2021, https://www.ar.admin.ch/en/beschaffung/ruestungspolitik-des-bundesrates/offset.html. (Accessed 01 June 2021).

Chi Thang Duong earned his Ph.D. degree in Computer Science at Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland. He received the Master and Bachelor degrees in computer science from EPFL and from Ho Chi Minh University of Technology, Vietnam. His research interests include graph analysis, network embedding, network alignment. He has publications in significant journals and conferences such as VLDBJ, SIGMOD, VLDB, and SIGIR.

Dimitri Percia David is an Assistant Professor of Data Science & Econometrics at the University of Applied Sciences (HES-SO Valais-Wallis). Previously, he was a postdoctoral researcher at the Information Science Institute of the University of Geneva (1st recipient of the EPFL Distinguished CYD Postdoctoral Fellowship). He applies data science and statistical learning methods to the field of technology mining. He earned his Ph.D. in Information Systems (HEC Lausanne, 2020), and worked 8+ years in the commodities-trading industry and ETH Zurich.

Ljiljana Dolamic is a Scientific Project Manager at Cyber-Defence Campus, armasuisse Science and Technology from 2019. She earned her Ph.D. in Information Retrieval from the University of Neuchatel in 2010. She was a scientific collaborator at ILCF, University of Neuchatel from 2016. She was a lecturer at Haute ecole de gestion Arc in 2018. She is a senior NLP expert with 7 years experience.

Alain Mermoud is the Head of Technology Monitoring and Forecasting at the Cyber-Defence Campus from armasuisse Science and Technology. His main research interests are in the area of emerging technologies, disruptive innovations, (cyber) threat intelligence and the economics of (cyber) security. In 2021, he was the General chair of the 16th International Conference on Critical Information Infrastructures Security (CRITIS 2021) hosted at EPFL. In 2019, he earned his Ph.D. in Information Systems from HEC Lausanne. Prior to that, he worked 5+ years in the banking industry and as lecturer at ETH Zurich.

Vincent Lenders is the founding Director of the Cyber-Defence Campus and the head of the Cyber Security and Data Science Department at armasuisse Science and Technology. He is also the co-founder and chairman of the executive boards of the OpenSky Network and Electrosense associations. He holds a Ph.D (2006) and a Master's (2001) degree in electrical engineering and information technology both from ETH Zürich. He was also postdoctoral research faculty in 2007 at Princeton University in the USA.

Karl Aberer is a full professor for Distributed Information Systems at EPFL, Switzerland. His research interests are on semantics in distributed information systems. He received his Ph.D. in mathematics in 1991 from the ETH Zurich. From 2012 to 2016 he was Vice-President of EPFL responsible for information systems. He is a co-founder of LinkAlong, a startup for document analytics. He is member of the editorial boards of VLDB Journal, ACM TAAS and WWW Journal and chairman of the ICDE Steering Committee.