Inferring multilingual domain-specific word embeddings from large document corpora

Received August 29, 2021, accepted September 26, 2021, date of publication October 5, 2021, date of current version October 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3118093

Inferring Multilingual Domain-Specific Word Embeddings From Large Document Corpora

LUCA CAGLIERO, (Member, IEEE), AND MORENO LA QUATRA, (Member, IEEE)
Dipartimento di Automatica e Informatica, Politecnico di Torino, 10129 Turin, Italy

Corresponding author: Luca Cagliero (luca.cagliero@polito.it)

This work was supported in part by the Smart-Data@PoliTO Center for Big Data and Machine Learning Technologies.

ABSTRACT The use of distributed vector representations of words in Natural Language Processing has become established. To tailor general-purpose vector spaces to the context under analysis, several domain adaptation techniques have been proposed. They all require sufficiently large document corpora tailored to the target domains. However, in several cross-lingual NLP domains both large enough domain-specific document corpora and pre-trained domain-specific word vectors are hard to find for languages other than English. This paper aims at tackling the aforesaid issue. It proposes a new methodology to automatically infer aligned domain-specific word embeddings for a target language on the basis of the general-purpose and domain-specific models available for a source language (typically, English). The proposed inference method relies on a two-step process, which first automatically identifies domain-specific words and then opportunistically reuses the non-linear space transformations applied to the word vectors of the source language in order to learn how to tailor the vector space of the target language to the domain of interest. The performance of the proposed method was validated via extrinsic evaluation by addressing the established word retrieval task. To this aim, a new benchmark multilingual dataset, derived from Wikipedia, has been released. The results confirmed the effectiveness and usability of the proposed approach.

INDEX TERMS Cross-lingual models, domain adaptation, natural language processing, word embeddings.

I. INTRODUCTION

In recent years, distributed vector representations of text have been widely applied to solve complex tasks in Natural Language Processing (NLP) such as sentiment analysis [1], machine translation [2], text categorization [3], and synonym prediction [4].

A pioneering word embedding model, namely Word2Vec, was proposed in [5]. The quality of its word-level text representations are impressive: it has shown to effectively capture most of the semantic word-level relationships in large document corpora. Later on, several new word-level encodings (e.g., FastText [6], GloVe [7]) and contextualized models (e.g., XLNet [8], ELMo [9], BERT [10]) have been proposed. The present study focuses on the Word2Vec model because, as discussed later on, it allows both word-level domain adaptation and multilingual alignment and still retains a high popularity level in several NLP applications [11].

Domain adaptation entails transforming high-dimensional vector spaces to specific domains [12]–[15]. The goal is to tailor the designed NLP solutions to specific application domains, such as energy [12], biology [15], and industry [14]. Within this scope, unsupervised domain adaptation techniques are particularly appealing, as they allow end-users to fine-tune a general-purpose model even in the absence of labeled data [13], [16].

Since the learning phase of the distributed representations of words relies on Deep Learning architectures, their computation requires (i) a sufficient large document corpora to learn robust data representations and (ii) an adequate computational power (e.g., ad hoc Graphical Processing Units) to accomplish the task in reasonable time. To overcome the above-mentioned issues, in the last decade the NLP community has released several pre-trained general-purpose multilingual models (see, for example, [7], [17]–[19]).

Multilingual document corpora are not only used to separately train language-specific embedding models, but also to align them in a unified latent space [19]. To this purpose, a bilingual lexicon is used to map the words of a source language (e.g., English) to the corresponding translations. Aligned word embedding models have been exploited

The associate editor coordinating the review of this manuscript and approving it for publication was Gianluigi Ciocca.

to effectively address cross-lingual NLP tasks, such as cross-lingual text classification [20], emotion lexicon induction [21], cross-lingual summarization [22]. As a drawback, in many cross-lingual NLP scenarios the use of aligned multilingual word embeddings is still limited by the lack of pre-trained domain-specific models for languages other than English. Currently, the greatest majority of pre-trained vectors were trained on general-purpose document corpora (e.g., Wikipedia). Just few domain-specific models are currently available and mostly for the English language (see Section V-A1). Moreover, for less spoken languages it can be very hard to retrieve a sufficiently large corpus of domain-specific documents. This calls for new approaches to automatically inferring aligned domain-specific multilingual word embeddings.

This paper presents a new inference method aimed at adapting the general-purpose Word2Vec vectors of a target language to its domain-specific version. The idea it is to rely on the underlying mapping between general-purpose and domain-specific word embeddings that is known for the English language. These aligned pre-trained models are either easy to retrieve or can be inferred thanks to abundance of English-written document corpora. In other words, the goal is to overcome the lack of domain-specific data and word vectors of the target language by exploiting data richness for the English language. Notice that the proposed approach can be easily extended to any application domain where the availability of data and word vectors of a specific language (not necessarily English) is prevailing.

The proposed method consists of a two-step inference process: first, it automatically identifies the sub-space of domain-specific words of the target language using a binary classifier. According to the domain under consideration, a word in the original space can either change its coordinates in the hyperspace if its relative position does not reflect the semantic similarity with its neighbor words in the domain-adapted space, or be invariant under domain adaptation if its general meaning is not influenced by the domain. The classification step discriminates between the two cases mentioned above. Hence, it allows us to tailor the next adaptation phase to a reduced word set (typically, one order of magnitude smaller than the original one) and, thus, to avoid introducing bias in the original model. Next, the proposed method infers the new position of each selected word in the domain-specific latent space. The latter step relies on a multivariate regression model trained on word vectors of the source language. The key idea is to learn and opportunistically reuse the (potentially non-linear) transformations that were previously applied by the multilingual embedding aligner to the words of the original language. Notably, the inference step is *aligner-agnostic*, i.e., it can be successfully applied whatever word embedding aligner was previously used on the aligned word vectors of the source language. As discussed later on, the proposed methodology is instrumental in addressing various cross-lingual NLP tasks (e.g., domain-specific text classification, text summarization).

Since, to the best of our knowledge, this is the first attempt to solve this particular issue, we crawled, prepared, and released a benchmark multilingual dataset tailored to our purposes. Benchmark data consist of (i) a set of document corpora retrieved from Wikipedia and written in seven different languages (i.e., Italian, English, French, Spanish, German, Arabic, Russian), (ii) the per-language word embeddings trained on general-purpose Wikipedia pages, (iii) a selection of terms related to specific domains (i.e., finance, technology, and medicine), (iv) the domain-specific, multilingual document corpora consisting of the term definitions on the basis of the Wikipedia interlanguage glossary, (iv) the per-language domain-specific embeddings.

To test the effectiveness and usability of the proposed method, we conducted an extrinsic evaluation of the model performance achieved on the word retrieval NLP task [23]. To this aim, we used the models trained on English documents as source vectors and separately tested the inferred domain-specific embeddings for the other languages (one by one) against the retrieved ground truth. We tested both linear and non-linear neural network-based regressors, relying on shallow and deep architectures. The results show that the models inferred using a deep fully-connected neural network model outperformed both general-purpose and linear models for most of the tested languages.

A. SUMMARY OF THE CONTRIBUTION

- To overcome the lack of domain-specific document corpora and pre-trained specialized models for less spoken languages, we study of the problem of domain adaptation in multilingual Word2Vec embeddings. This work is, to our best knowledge, the first attempt to address the aforesaid research issue.
- We propose a two-step inference process based on (i) automatic identification of domain-specific words and (ii) supervised inference of the new word vectors in the domain-specific hyperspace of the target language.
- We release a new benchmark multilingual dataset tailored to the task under consideration. To the best of our knowledge, this is first benchmark including general-purpose, multi-domain, and multilingual data and aligned word vectors at the same time.

The rest of the paper is organized as follows. Section II presents the preliminary results achieved in two practical NLP use cases. Section III overviews the related works and discusses the position of the present paper in the related literature. Section IV thoroughly describes the proposed methodology. Section V summarizes the results of the empirical evaluation, whereas Section VI draws conclusions and discusses the future research agenda.

II. MOTIVATING EXAMPLES

We report and qualitatively describe here the preliminary outcomes achieved by adopting the proposed method to address two well-known NLP tasks, i.e., word analogy [24]

and retrieval [23]. The respective results are summarized in Tables 1 and 2, where *Base* indicates the outcomes produced by exploiting the general-purpose models, whereas *Domain* denotes the outcomes produced by the inferred models tailored to the technology domain (assuming that a sufficient amount of domain-specific data is not available to directly train the domain-specific model).

A. WORD ANALOGY TASK

The word analogy task entails answering analogical questions like *man is to king as woman is to?* by specifying the most appropriate word (e.g., *queen*). Word embeddings have relevantly simplified and improved the performance of the NLP approaches used to tackle the above-mentioned task. Specifically, in [5] the authors showed that Word2Vec embedding exhibits seemingly linear behaviour. The embeddings of the analogy *woman is to queen as man to king* approximately describe a parallelogram [25], even if the model is not specifically trained to address such a task. Hence, given the vector representation of words *man*, *king*, and *woman* in the hyperspace, the analogical questions *man is to king as woman is to?* can be solved by simply computing a linear combination of vectors in the hyperspace ($v_{king} - v_{man} + v_{woman}$).

For each analogical question, Table 1 reports the top-5 nearest neighbor words in the vector space corresponding to the resulting vector. The aim is twofold: (i) test the ability of the models to retrieve appropriate words at the top of the rank and (ii) compare the rank produced by the general-purpose model with those achieved by the domain-specific ones. The latter are expected to produce more pertinent answers questions related to the technological domain. The results confirmed the expectation for all the tested languages.

B. MOST SIMILAR WORD RETRIEVAL TASK

The word analogy task entails answering a query by retrieving the most similar words. The goal is to evaluate the ability of domain-specific models to better capture the semantic relationships among words belonging to the technological domain.

Table 2 summarizes the achieved results, which highlight the specialization of the inferred model. For example, given the query *memoria* (i.e., the Italian word for *memory*) it retrieves words like *usb* rather than *ricordo* or *commerazione*, which are the Italian translation of *recollection* and *remembrance*, respectively.

A quantitative evaluation of the performance of the proposed method in solving this particular task is given in Section V.

III. RELATED WORK

The main goal of word embedding methods is to organize words into a Poincaré hyperspace such that their distance reflects their semantic similarity [26]. To achieve this goal, the learning process relies on the distributional hypothesis. The rationale behind such an hypothesis is that linguistic items occurring within the same domain likely have similar meanings [27]. Hereafter we will separately present (i) the most relevant word embedding models, (ii) the studies aimed at tailoring general-purposes models to specific domains, (iii) the strategies used to align embeddings in multilingual contexts, and (iv) the efforts made in contextualized embeddings. Finally, we will clarify the position of the present work in the state-of-the-art literature.

A. WORD EMBEDDING MODELS

Training vector representations of text using neural networks was first proposed by Bengio *et al.* [28], whose main goal was to learn a probabilistic language model. A pioneering work in this field was presented in [5]. Given a large training corpus, the authors proposed an effective and efficient neural network-based approach (namely Word2Vec) to learning word embedding based on a sliding window strategy. The indisputable success of the Word2Vec model in supporting several NLP tasks has fostered a huge body of work on learning vector space models. For example, FastText [6] extended the Word2Vec model by encoding also sub-words. This alleviates the Out-Of-Vocabulary problem since the network can infer the embedding of a new word by combining the vector representations of the n-grams that compose it. GloVe [7] and MWE [29] inferred word vector representations based not only on the local context of a word, but also on global information reported in a word co-occurrence matrix. The present study focuses on Word2Vec. Notice that, unlike FastText, Glove, and MWE, Word2Vec supports both word-level domain adaptation and multilingual word vector alignment.

B. DOMAIN ADAPTATION

Word embeddings may differ from one domain to another due to lexical and semantic text variations. Hence, their performance have shown to be strongly dependent on the training corpus [30]. To capture domain specificity a relevant research effort has been devoted to fine-tuning general-purpose vector spaces to capture the peculiarities of specific domains. For example, the method presented in [31] focuses on capturing the word polysemy in different contexts based on topic modeling, whereas in [32] a meta-learner is used to expand the in-domain corpus by exploiting the corpora from a set of past related domains.

Unsupervised domain adaptation approaches (e.g., [33]) often rely on ad hoc heuristics to identify *pivot words*, i.e., words that are frequently used in a specific domain. Domain adaptation is crucial to successfully employ the embedding model in specific application areas such as finance and healthcare [13]. For example, in [12] and [15], [34] the authors empirically demonstrated how document corpora respectively ranging over oil/gas and biomedical domains can be exploited to improve the quality of word embeddings. In [14] the authors proposed an architecture aimed at adapting general-purpose word embeddings using industry-specific data in order to improve document classifier performance. The benefits of using specialized word

TABLE 1. Qualitative comparison between the general-purpose model (Base) and the inferred domain-specific (Domain) model. Word analogy task. Domain: technology.

Language	Analogical question	Model	1st	2nd	3rd	4th	5th
English	"Computer" - "Work" + "Game"?	Base	smileboom	robocraft	corecell	mexond	computers
		Domain	consoles	snes	odallus	joystick	batterup
Italian	"Computer" - "Lavoro" + "Gioco"?	Base	videogioco	commodore	amiga	arkanoid	superhot
		Domain	arcade	videogioco	wii	videogame	multigiocatore
French	"Ordinateur" - "Travail" + "Jeu"?	Base	vectrex	chessmaster	mindlink	croteam	multijoueur
		Domain	croteam	vidéo	evoland	vectrex	pixeljunk
Spanish	"Computadora" - "Trabajo" + "Juego"?	Base	ordenador	gyromite	computadora	intellivision	multijugadores
		Domain	consola	videojuego	jugabilidad	multijugador	famicom
German	"Computer" - "Arbeit" + "Spiel"?	Base	gamepad	jetpac	vectrex	handyspiel	computerspiel
		Domain	denkspiel	spielkonsole	jetpac	gamepad	eingabegerät

TABLE 2. Qualitative comparison between the general-purpose model and the inferred domain-specific model. Most similar word retrieval task. Domain: technology.

Language	Term	Model	1st	2nd	3rd	4th	5th
English	memory	Base	memories	legacy	dedicated	nondeclarative	autoassociative
		Domain	procedural	nvdimm	memories	kilobytes	nvram
Italian	memoria	Base	ricordo	tramandarne	commemorazione	perpetuarne	eidetica
		Domain	memorizzazione	pendrive	eeprom	usb	vram
French	mémoire	Base	mémorielle	mémorialisation	souvenir	dpram	mémoires
		Domain	eprom	eeprom	dram	adressable	cache
Spanish	memoria	Base	memoría	recuerdo	paginada	remhi	eternizar
		Domain	memoría	caché	eeprom	paginada	pendrives
German	speicher	Base	speichers	hauptspeicher	arbeitsspeicher	strataflash	hornscher
		Domain	arbeitsspeicher	hauptspeicher	festplatte	mbyte	massenspeicher

embedding models have been demonstrated in languages other than English as well [35].

C. BILINGUAL EMBEDDING ALIGNMENT

Several studies have investigated the alignment between pairs of embedding models (namely, the *source* and *target* models). The goal is to map words of a source language to the corresponding ones of the target language. This is particularly useful for addressing automated machine translation [36]. Unsupervised approaches (e.g., [37], [38]) focused on learning a transformation from the source to the target by assuming an empirical distribution in the embedding models, whereas supervised strategies (e.g., [19], [39], [40]) relied on bilingual lexicons.

D. CONTEXTUALIZED EMBEDDINGS

Contextualized embeddings are vector representations of text where a target word's embedding can change depending on the context in which it appears [8], [9], [41], [42]. Unlike Word2Vec, FastText, and Glove they rely on a dynamic representation for each word. Therefore, by construction, they are unsuitable for generating multilingual word-level vector alignments.

E. POSITION OF THE PRESENT WORK IN THE STATE OF THE ART

- This work focuses on Word2Vec embeddings [5]. FastText [6] is not applicable because it relies on sub-words compositionality thus it can be aligned only for static embedding models. GloVe [7] cannot be used since it is based on the corpus' overall word co-occurrence statistics from a single corpus known only at initial training time.
- The aim is to adapt multilingual general-purpose word embeddings to a specific domain to overcome the lack of domain-specific data. Hence, it is a combination of the domain adaptation and bilingual alignment tasks.
- The aim is *not* to propose new ad hoc solutions separately for the domain adaptation and supervised embedding alignment tasks.
- The use of contextualized embeddings is out of scope of the present work and will be addressed as future work (see Section VI).

IV. PROPOSED METHODOLOGY

Let $\mathcal{L}$ be a set of languages and let V_l be the vocabulary of words of a language $l \in \mathcal{L}$. We assume that we have sets of word embeddings E_l ($l \in \mathcal{L}$) trained independently on monolingual data. We differentiate between *general-purpose* embeddings E_l^G, i.e., word embeddings trained on multi-domain, general-interest document corpora such as the whole Wikipedia corpus, and *domain-specific* embeddings E_l^δ, which are specialized using document corpora tailored to a specific domain δ.

Algorithm 1 reports the main steps of the proposed methodology. A graphical sketch of key phases is depicted in Figure 1. The procedure takes as input the general-purpose and domain-specific document corpora for the source language as well as the general-purpose corpus for one or more target languages. The expected outcome is to infer domain-specific word embedding models separately for each target language. Once all general-purpose embedding models are trained, the model corresponding to the source language is fine-tuned by exploiting a domain-specific corpus

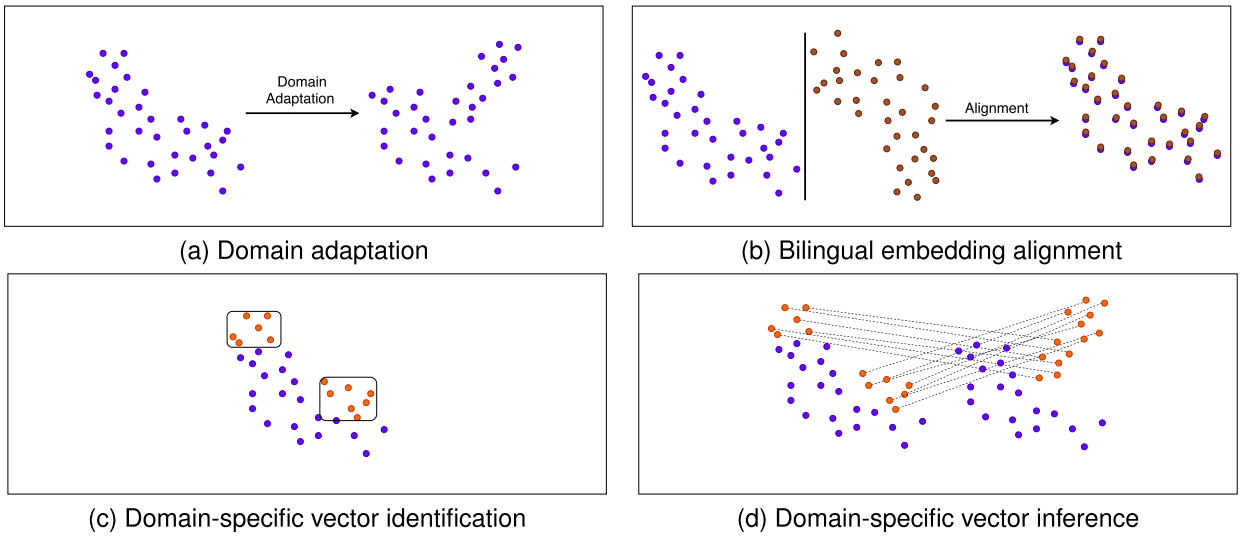

FIGURE 1. Sketch of the main methodology steps.

(see Figure 1a). In the current implementation of the proposed method, both model training and domain adaptation rely on Word2Vec [5]. However, the embedding method can be straightforwardly substituted with any other word-level embedding that allows domain-adaptive fine-tuning. Then, the general-purpose models for the target languages are all aligned to the corresponding model for the source language by adopting the supervised approach proposed by [19] (see Figure 1b). As in the previous step, different bilingual alignment strategies can be easily integrated as well. Next, to infer domain-specific embeddings for the target languages it is first necessary to discriminate between words specific to the target domain and not. To this aim, a binary classifier is trained on the source language models to predict which words in the general-purpose model of each target language are likely to be specific to the target domain (see Figure 1c). For the subset of words of the target language that are labelled as domain-specific ($\mathbb{V}_l^{true}$), new vectors are inferred by using a regression model (see Figure 1d). The regression step learns from the embeddings available in the source language the mapping between word vectors of the general-purpose and domain-specific models. The mapping is opportunistically reused to infer new word vectors for the target languages. Finally, the newly inferred vectors are joined with the word vectors labeled as not domain-specific ($\mathbb{V}_l^{true}$) to compose the complete domain-specific embeddings for the target languages E_t^{δ}.

A more thorough description of each step is given in Algorithm 1.

A. DOMAIN ADAPTATION

For each language $l \in \mathcal{L}$ the domain adaptation phase takes as input the general-purpose embedding E_l^G and the domain-specific corpora $\mathbf{D}^{\delta}$. It generates the corresponding domain-specific embedding E_l^{δ} (see Figure 1a).

This phase entails fine-tuning the general-purpose model by shifting the vectors of domain-specific words in order to better capture their context-specific semantic meaning. The key idea is to specialize the general-purpose model for the source language (typically, English) for which a sufficiently large amount of domain-specific data are available. Such a specialized model will be opportunistically re-used to infer the mapping between general-purpose and domain-specific models for the target languages.

Notice that, at this stage, pretrained general-purpose models (e.g., [17]) can be exploited to avoid retraining the vector representations of the source text from scratch. Despite a number of open-source projects having released general-purpose models (more details are given in Section V-A), only few of them include domain-specific data and models and mostly for a limited number of languages. The latter evidence inspired our research.

B. BILINGUAL EMBEDDING ALIGNMENT

Let $\langle l_s, l_t \rangle$ be a pair of *source* and *target* languages. Each word w^i in the vocabulary of the source language (respectively target language) is associated with a vector $\mathbf{x}^i \in \mathbb{R}$. To align the two corresponding embeddings E_{l_s} and E_{l_t} we exploit an initial bilingual lexicon, of size d, that maps each word w_s^i of the source language to the corresponding translation w_t^i of the target language. The bilingual alignment step entails extending the lexicon to all source words in V_{l_s} that are not present in the initial lexicon so that all word vectors E_{l_s} have an explicit mapping to E_{l_t} (see Figure 1b). State-of-the-art alignment methodologies leverage bilingual lexicons to optimizes a retrieval criterion able to generalize on the full vocabulary learning a source-to-target alignment function.

In our context, we consider as source language the one for which both general-purpose and domain-specific corpora are given (typically, English). The target language is the language for which only general-purpose document corpora

Algorithm 1: Proposed Methodology

Result: E_t^δ: Domain-specific word embedding in the target languages ($\forall\, l \in \mathcal{L}$)

Input : $\mathbf{D}_s^G$: General-purpose document corpus in the source language;
$\mathbf{D}_s^\delta$: Domain-specific document corpus in the source language;
$\mathcal{L}$: target languages;
$\mathbf{D}_t^G$: General-purpose document corpus in the target language ($\forall\, l \in \mathcal{L}$)

```
/* Train general-purpose embeddings */
```
$E_s^G \leftarrow$ Word2Vec($\mathbf{D}_s^G$)
foreach $l \in \mathcal{L}_t$ **do**
  $E_l^G \leftarrow$ Word2Vec($\mathbf{D}_l^G$)
end
```
/* Source language embedding adaptation */
/* (See Figure 1a) */
```
$E_s^\delta \leftarrow$ Fine-Tuning(E_s^G, $\mathbf{D}_s^\delta$)
```
/* Target-to-source embedding alignment */
/* (See Figure 1b) */
```
foreach $l \in \mathcal{L}_t$ **do**
  $\mathbb{W}_l^* \leftarrow$ RCSLS(E_l^G,E_s^G)
end
```
/* Domain-specific vector training */
```
$\mathcal{C} \leftarrow$ Classifier-training(E_s^δ, E_s^G)
$\mathcal{R} \leftarrow$ Regressor-training(E_s^δ, E_s^G)
```
/* Domain-specific vector identification */
/* (See Figure 1c) */
```
foreach $l \in \mathcal{L}_t$ **do**
  $\langle \mathbb{V}_l^{true}, \mathbb{V}_l^{false} \rangle \leftarrow$ Apply-Classifier($\mathcal{C}$,$\mathbb{W}_l^*$)
end
```
/* Domain-specific vector inference */
/* (See Figure 1d) */
```
foreach $l \in \mathcal{L}_t$ **do**
  $\mathbb{E}_l^{\delta,NEW} \leftarrow$ Apply-Regressor($\mathcal{R}$,$\mathbb{V}_l^{true}$)
end
foreach $l \in \mathcal{L}_t$ **do**
  $E_t^\delta = \mathbb{E}_l^{\delta,NEW} \cup \mathbb{V}_l^{false}$
end
return E_t^δ: Domain-specific word embedding in the target languages ($\forall\, l \in \mathcal{L}$).

are currently available but there is a need to learn domain-specific word embeddings.

Let $\mathbb{M}_s$ and $\mathbb{M}_t$ be the matrices of real numbers respectively containing the words embeddings in E_{l_s} and E_{l_t} of the words in the initial lexicon. Bilingual embedding alignment entails learning a linear mapping $\mathbf{W}$ between the source and target hyperspaces so that the discrepancy between the corresponding word vectors is minimized.

$$\mathbf{W}^* = arg_{\mathbb{W}\in\mathbb{R}^{d\times d}} min \left\| \mathbf{W}\mathbf{x_i^t} - \mathbf{y_j^s} \right\|_2^2$$

where $\mathbf{x_i^t}$ and $\mathbf{y_j^s}$ are mapped word vectors in the source and target spaces and $\left\| \mathbf{x_i^t} - \mathbf{y_j^s} \right\|$ is the square loss function to be minimized.

To align bilingual word embeddings we exploited the supervised approach proposed in [19] by considering, as initial bilingual lexicons, the ones released by [39].[1]

C. DOMAIN-SPECIFIC VECTOR IDENTIFICATION

The classification step (depicted in Figure 1c) aims at classifying each word vector $\mathbf{x}^i$ belonging to the general-purpose embedding $E_{l_t}^G$ for the target language l_t as follows.

$$l(\mathbf{x}^i) = \begin{cases} true, & \text{if } w_i \text{ is likely to be domain-specific} \\ false, & \text{otherwise} \end{cases}$$

To accomplish this task we study the correlation between general-purpose and domain-specific word vectors E_s^δ and E_s^G in the source language. The idea behind it is to rely on the empirical evidence from the domain adaptation process previously applied to the source language. Specifically, the word vector shifts that would be produced by domain adaptation for the target language are expected to reflect, to a good approximation, those observed for the source language. Hence, similar word vectors are likely to show similar shifts in the adaptation phase. The word-level prediction model can be formulated as the following boolean function $\mathbf{f}$

$$l(\mathbf{x}^i) = \mathbf{f}(\mathbf{x}^i, E_s^\delta, E_s^G)$$

D. DOMAIN-SPECIFIC VECTOR INFERENCE

This step builds the domain-specific embeddings E_t^δ. They consist of (i) the vectors of domain-specific words (i.e., the words labeled as true at the previous step), which are likely to change with respect to the corresponding vector in E_t^G, and (ii) vectors of not domain-specific words (i.e., the words labeled as false), which are invariant under domain adaptation as their semantic meaning is unlikely to be influenced by the domain under consideration. To estimate the domain-specific vectors we infer the position of the type-(i) vectors using a regression model, whereas we approximate the type-(ii) vectors as those already available in the general-purpose model (i.e., we assume that domain adaptation does not yield any type-(ii) vector shift in the hyperspace).

Analogously to what previously done for domain-specific vector identification, we learn how to shift word vectors for the target language by studying the correlations between general-purpose and domain-specific word vectors E_s^δ and E_s^G in the source language. At this stage, we predict the exact

[1] https://github.com/facebookresearch/MUSE (latest access: June 2021)

values of each element of the new vector by learning the following regressor $\mathbf{r}$:

$$\mathbf{x}^j = \mathbf{r}(\mathbf{x}^i, E_s^\delta, E_s^G)$$

where $\mathbf{x}^i$ is the vector associated with word w_t^i in the general-purpose model, whereas $\mathbf{x}^j$ is the vector associated with the same word in the domain-specific model (after the eventual shift due to domain adaptation).

V. EXPERIMENTAL RESULTS

We summarize here the outcomes of the empirical analysis carried out on the document corpora retrieved from Wikipedia. Specifically, Section V-A describes the newly released benchmark dataset, Sections V-B and V-C formalize the addressed NLP task and the tested models, respectively. Section V-D reports the outcomes of the performance comparison. Section V-E analyzes the effect of the system parameters.

The experiments were run on machine equipped with 32GB of RAM, Intel Xeon E5-2680 CPU and Nvidia Tesla K40 GPU.[2]

The computational time required by the overall process of domain-specific model inference (including both classification and regression) was quite variable across languages. It ranged from 51 seconds (Arabic language) to 175 seconds (German language).

A. BENCHMARK DATASET

The lack of open multingual datasets that fit for our purposes prompted us to crawl, prepare, and release a new benchmark dataset, namely AMED (Adapting Multilingual word Embeddings to specific Domains).[3]

The AMED benchmark dataset consists of a set of multilingual document corpora retrieved from Wikipedia and ranging over different topics. The Wikipedia online encyclopedia is a common source of data to learn word representations, as it is available in many languages [17]. More specifically, it includes

1) The full Wikipedia dump crawled in November 2020 separately for each of the following languages: Italian, English, French, Spanish, German, Arabic, Russian.
2) The general-purpose word embedding models trained on the per-language Wikipedia dumps at Point (1).
3) For a subset of domains (i.e., medicine, technology, finance), the lists of most representative terms in the Wikipedia glossary[4] translated in all the languages considered at Point (1).
4) The multilingual document corpora consisting of the definitions of the selected Wikipedia terms retrieved at Point (3). Definitions are given in all the languages considered at Point (1).
5) The domain-specific word embedding models adapted to the domains specified at Point (3) by using the multilingual document corpora selected at Point (4).

The multilingual document corpora used to train the general-purpose models were retrieved from the latest dump of the language-specific wikipedia encyclopedia.[5] Domains at Point (3) were selected among the most common categories in the English Wikipedia dump (e.g., https://en.wikipedia.org/wiki/Category:Finance). Glossary terms at Point (4) were extracted by considering the corresponding glossary sub-categories. The domain-specific documents at Point (4) were retrieved by first querying the Wikipedia glossary in English through the PetScan tool[6] and then by following the corresponding Wikipedia interlanguage links[7] in order to retrieve consistent documents across different languages.

Table 3 summarizes the main data characteristics. As one can clearly deduce by the reported statistic, the English corpus is six times larger than those of available in the other languages (6 Millions vs. 1 Million). Furthermore, the number of domain-specific documents tailored to a single domain is significantly smaller (three order of magnitude lower). This reinforces the motivations behind our research: in contexts where retrieving a sufficiently large corpora written in languages other than English is challenging (e.g., summarization of patents or technical reports, conversation agents for technical support, multilingual search engines) training domain-specific models would be challenging. Finally, the characteristics of the textual definitions are rather diversified across languages (e.g., definitions in Russian contain approximately half of the words than those in all the other languages).

1) COMPARISON WITH EXISTING BENCHMARKS

Other researchers have previously released large textual corpora and word embedding models along with the open source implementations of their research projects. For example, in [5] the authors released English word embedding trained on Google News; in [7] released English models trained on Wikipedia, Gigaword and Common Crawl. In [18] the authors released general-purpose word embeddings trained for 100 languages based on Wikipedia data. [6] and [17] respectively released FastText and Word2Vec word embeddings for 44 and 157 languages using Wikipedia and data from the common crawl project. However, to the best of our knowledge, a benchmark dataset consisting of both general-purpose and domain-specific embeddings in various domains and languages has not been presented in literature yet.

[2]Computational resources for Deep Network training were provided by HPC@POLITO, a project of Academic Computing within the Department of Control and Computer Engineering at the Politecnico di Torino.

[3]https://github.com/MorenoLaQuatra/AMED (latest access: August 2021)

[4]https://en.wikipedia.org/wiki/Wikipedia:Glossary (latest access: June 2021)

[5]We crawled data from the following URLs: https://dumps.wikimedia.org/XXwiki/latest/, where XX must be replaced with the ISO 639-1 language code.

[6]https://petscan.wmflabs.org/ (latest access: June 2021)

[7]https://en.wikipedia.org/wiki/Help:Interlanguage_links (latest access: June 2021)

TABLE 3. Characteristics of the AMED benchmark dataset.

Domain	Language	Corpora			Terms		Definitions
		# Docs.	# words	# unique words	# queries	Avg. # words per term	Avg. # words per definition
Medicine	English	17,243	8,588,618	678,972	1139	2.61	18.23
	Italian	244	131,440	41,275	213	1.89	18.46
	French	311	224,378	57,458	281	2.15	17.62
	Spanish	309	180,276	52,737	256	2.04	18.03
	German	364	212,621	76,039	337	1.45	16.61
	Arabic	602	478,602	73,569	378	2.68	19.63
	Russian	292	253,431	57,718	227	1.9	10.04
Technology	English	21,427	12,583,640	970,939	2153	2.75	18.29
	Italian	333	219,742	61,766	304	2.08	19.15
	French	449	387,037	81,637	405	2.2	18.95
	Spanish	485	313,217	80,846	441	2.17	19.09
	German	481	276,434	98,367	447	1.75	18.82
	Arabic	466	390,728	71,131	292	2.69	19.37
	Russian	408	471,315	100,700	368	2.28	10.82
Finance	English	6,674	3,724,202	317,290	1807	2.76	18.7
	Italian	231	120,068	31,159	204	2.51	18.62
	French	391	244,447	47,369	341	2.85	18.01
	Spanish	303	148,900	37,546	262	2.68	18.71
	German	443	253,860	71,762	400	1.64	16.21
	Arabic	676	561,172	44,920	199	2.66	19.63
	Russian	711	517,862	48,754	312	2.39	11.4

B. WORD RETRIEVAL TASK

The retrieval task is known since long ago [23] and has been largely addressed by the NLP community (e.g., [43]–[45]).

To extrinsically evaluate the quality of the inferred models we formulated the retrieval task on the benchmark dataset as follows: *given a Wikipedia term retrieve the keyphrases in the corresponding glossary definition*. Since this work focuses on *word* embeddings, we applied the following data preparation steps:

1) For each term in the multilingual Wikipedia glossaries[8] retrieve the title of the corresponding Wikipedia page.
2) Extract the set of words occurring in the title (excluding the stopwords).
3) **Term** $\leftarrow w_1^T, w_2^T, \ldots, w_n^T$
4) Summarize the Wikipedia page using the top-2 sentences in the document.
5) Extract the set of words occurring in the keyphrases (except for the stopwords).
6) **Definition** $\leftarrow w_1^D, w_2^D, \ldots, w_m^D$

To our purposes, we reformulate the word retrieval task as follows: given a *term* retrieve the words in the *definition*.

1) EXTRINSIC EVALUATORS

We extrinsically evaluate model effectiveness in addressing the word retrieval task in terms of Precision, Recall and F1-Measure [46]. The aforesaid measures are established in Information Retrieval [23].

For each term T we first retrieve a ranked list of words *Ret*. Then, we evaluate the pertinence of the retrieved words placed at the top of the ranking to the description as follows.

$$P@K = \frac{Ret_K}{K}$$
$$R@K = \frac{Ret_K}{|D|}$$
$$F@K = 2 \cdot \frac{P@K \cdot R@K}{P@K + R@K}$$

where K is the target number of top ranked words to retrieve, Ret_K is the number of words in the top-K of *Ret* that were actually retrieved from the description D, and $|D|$ is the total number of words in the description.

Precision is the percentage of correctly retrieved words over the total number of retrieved words, recall is the percentage of correctly retrieved words over the total number of description words to be retrieved, whereas F1-measure is the harmonic average of precision and recall.

The aforesaid measures will be averaged over all the analyzed terms in order to get a unique quality score per model. Notice that the number of words in the definition approximately doubles the number K of words to retrieve (see Table 3). The only exception is the Russian language, where the two aforesaid counts are approximately equal.

C. EMBEDDING MODELS

We tested the following multilingual embedding models:

- **General-Purpose (GP)**: the general-purpose Word2Vec embedding model trained on the target language.
- **Ground Truth (GT)**: the domain-specific Word2Vec embedding model obtained by adapting the general-purpose model for the target language using all the available domain-specific corpora written in the target language.
- **Linear Inference Model (LIM)**: the word embedding model inferred from the general-purpose one for the target language using the proposed method. The inference relies on linear classifiers and regressors.

[8]Glossary examples. English: https://en.wikipedia.org/wiki/Wikipedia:Glossary Italian: https://it.wikipedia.org/wiki/Aiuto:Glossario (latest access: June 2021)

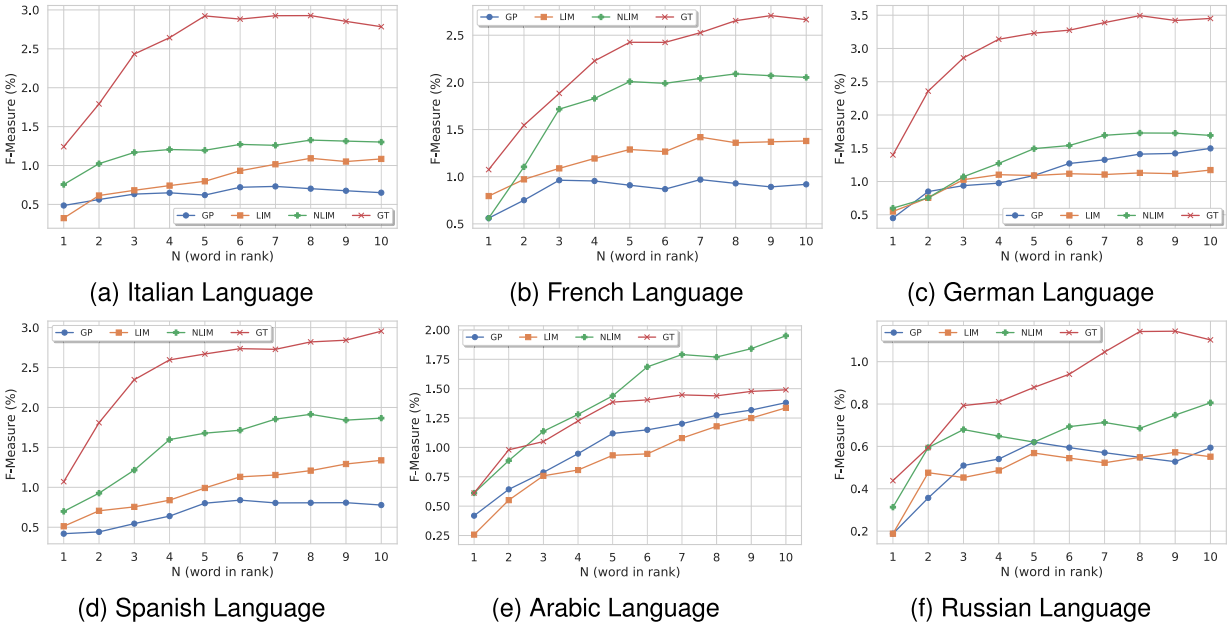

FIGURE 2. Comparison between general-purpose, linear and non-linear models, and ground truth in terms of F1-measure. Medical domain.

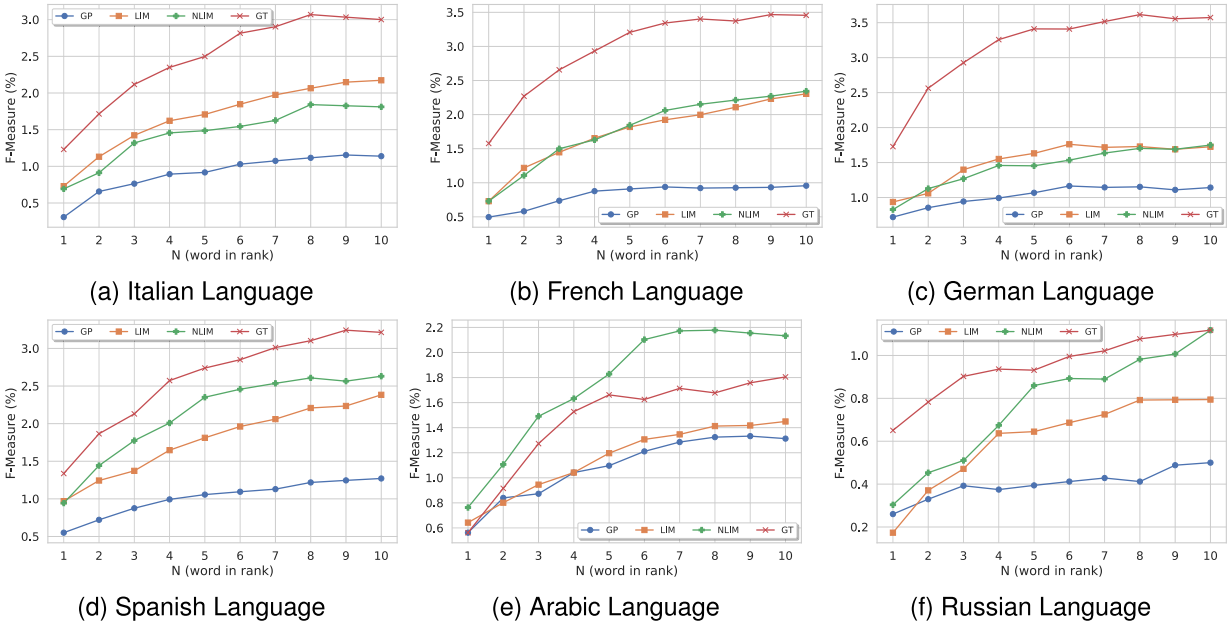

FIGURE 3. Comparison between general-purpose, Linear and Non-linear models, and Ground Truth in terms of F1-measure. Technological domain.

- **Non-Linear Inference Model (NLIM)**: the word embedding model inferred from the general-purpose one for the target language using the proposed method. The inference process relies on non-linear classifiers and regressors.

Both word embedding training and fine-tuning phases were performed using the Gensim library [47]. The GP model will be used as a reference to get a lower-bound estimate of the performance, as domain-specific models are expected to perform better the general-purpose ones. Conversely, the performance of the GT model will be considered as an upper bound estimate since the proposed inference method is assumed not to take advantage of domain-specific data in the target language. The closer the extrinsic evaluation score to the GT's ones, the better the result.

LIM is the proposed inference method, where both classification and regression step rely on linear predictive models. As linear models we considered Linear Regressor and Support Vector Classifier available in the the SciKit-Learn library [48]. NLIM is the variant of the proposed inference

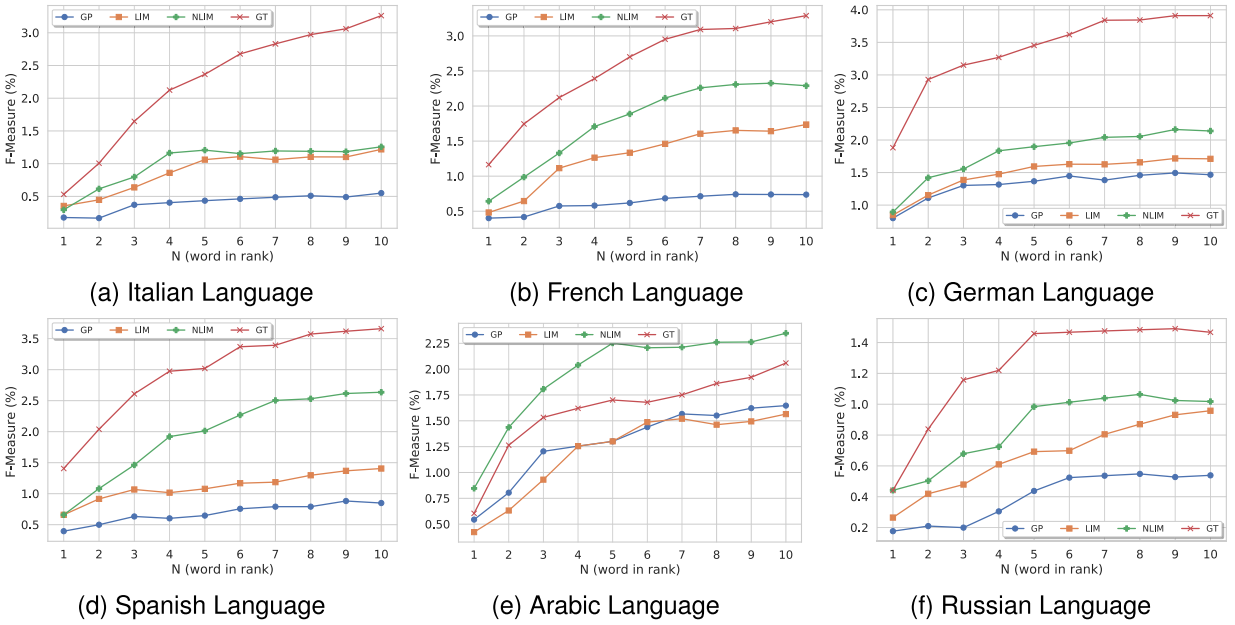

FIGURE 4. Comparison between general-purpose, linear and non-linear models, and ground truth in terms of F1-measure. Financial domain.

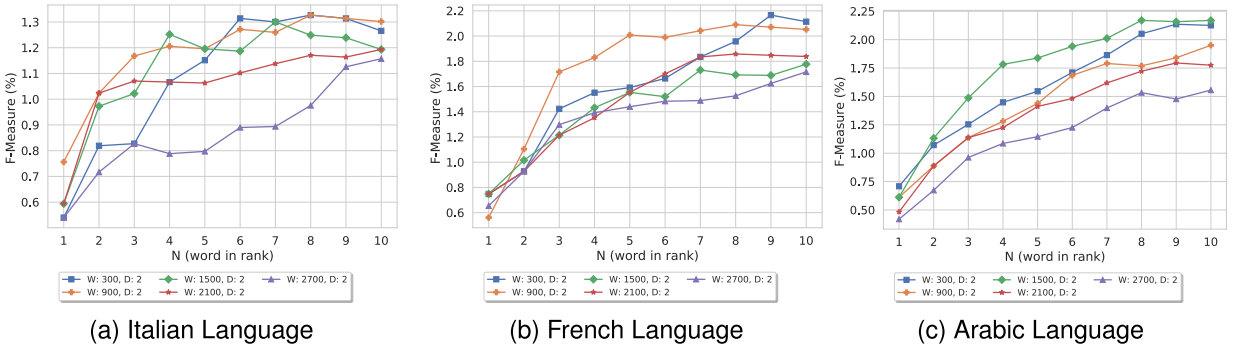

FIGURE 5. Effect of the network width. 2-Layer fully connected neural network model. Medical domain.

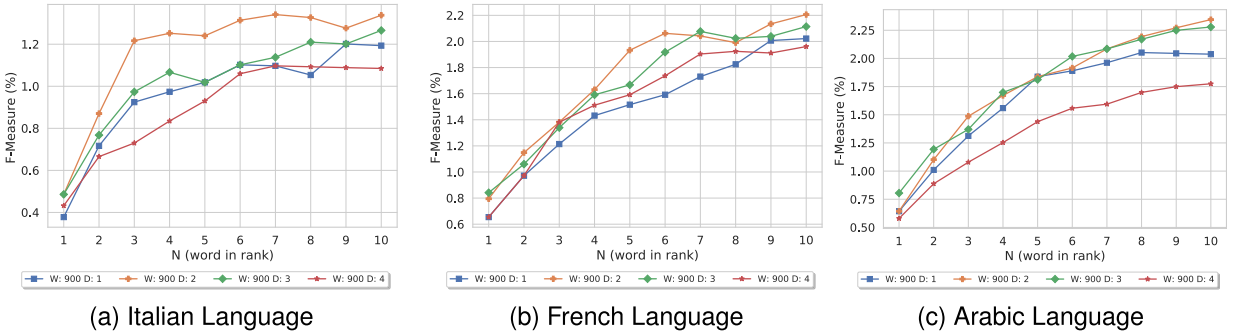

FIGURE 6. Effect of the network depth. Non-Linear fully connected neural network model. Medical domain.

model, where both steps potentially rely on non-linear predictions. The comparison between LIM and NLIM is aimed at understanding to extent to which non-linear predictors could enhance model performance compared to simpler (linear) ones. In NLIM we explored the use of deep learning neural network-based models as well. Specifically, as non-linear models we relied on a fully connected neural networks (MultiLayer Perceptron with ReLU activation function) and explored both shallow and deep versions of the network architecture (more details are given in Section V-E).

D. PERFORMANCE COMPARISON

Figures 2-4 plot the per-language F1-measure scores (with K between 1 and 10) achieved by Baseline, Linear, Non-Linear,

TABLE 4. Comparison between general-purpose, LInear and Non-LInear models, and ground truth.

Language	K	General-Purpose				LINear				Non-LINear				Ground Truth		
		Pr (%)	Rec (%)	F1 (%)	ΔF1vs.G.T. (%)	Pr (%)	Rec (%)	F1 (%)	ΔF1vs.G.T. (%)	Pr (%)	Rec (%)	F1 (%)	ΔF1vs.G.T. (%)	Pr (%)	Rec (%)	F1 (%)
Medicine																
Italian	3	2.13	0.37	0.63	25.93	2.3	0.4	0.68	27.98	3.94	0.69	1.17	**48.15**	8.21	1.43	2.43
	7	1.27	0.51	0.73	24.91	1.76	0.71	1.02	34.81	2.18	0.89	1.26	**43.0**	5.07	2.06	2.93
	10	0.89	0.51	0.65	23.38	1.48	0.86	1.08	38.85	1.77	1.03	1.3	**46.76**	3.79	2.2	2.78
German	3	2.94	0.56	0.94	32.87	3.22	0.61	1.03	36.01	3.36	0.64	1.07	**37.41**	8.96	1.7	2.86
	7	2.16	0.96	1.33	39.23	1.8	0.8	1.11	32.74	2.76	1.22	1.69	**49.85**	5.52	2.44	3.39
	10	1.93	1.22	1.5	43.48	1.51	0.96	1.17	33.91	2.18	1.38	1.69	**48.99**	4.45	2.82	3.45
French	3	3.07	0.57	0.96	51.06	3.47	0.65	1.09	57.98	5.47	1.02	1.72	**91.49**	6.0	1.12	1.88
	7	1.6	0.7	0.97	38.34	2.34	1.02	1.42	56.13	3.37	1.46	2.04	**80.63**	4.17	1.81	2.53
	10	1.2	0.74	0.92	34.46	1.8	1.12	1.38	51.69	2.68	1.66	2.05	**76.78**	3.48	2.16	2.67
Spanish	3	1.83	0.32	0.55	23.4	2.53	0.44	0.75	31.91	4.08	0.71	1.22	**51.91**	7.88	1.38	2.35
	7	1.39	0.57	0.8	29.3	1.99	0.81	1.15	42.12	3.19	1.31	1.85	**67.77**	4.7	1.92	2.73
	10	1.05	0.62	0.78	26.35	1.81	1.06	1.34	45.27	2.53	1.48	1.87	**63.18**	4.01	2.34	2.96
Arabic	3	2.78	0.46	0.79	75.24	2.67	0.44	0.76	72.38	4.01	0.66	1.14	**108.57**	3.7	0.61	1.05
	7	2.16	0.83	1.2	82.76	1.94	0.75	1.08	74.48	3.22	1.24	1.79	**123.45**	2.6	1.0	1.45
	10	1.94	1.07	1.38	92.62	1.88	1.04	1.34	89.93	2.75	1.51	1.95	**130.87**	2.1	1.16	1.49
Russian	3	1.78	0.3	0.51	64.56	1.58	0.26	0.45	56.96	2.37	0.4	0.68	**86.08**	2.76	0.46	0.79
	7	1.01	0.4	0.57	54.29	0.93	0.36	0.52	49.52	1.27	0.5	0.71	**67.62**	1.86	0.73	1.05
	10	0.83	0.46	0.59	53.64	0.77	0.43	0.55	50.0	1.12	0.63	0.81	**73.64**	1.54	0.86	1.1
Technology																
Italian	3	2.61	0.45	0.76	35.85	4.86	0.83	1.42	**66.98**	4.51	0.77	1.32	62.26	7.24	1.24	2.12
	7	1.88	0.75	1.07	36.9	3.46	1.38	1.97	**67.93**	2.85	1.14	1.63	56.21	5.08	2.03	2.9
	10	1.57	0.89	1.14	38.0	2.99	1.71	2.17	**72.33**	2.49	1.42	1.81	60.33	4.13	2.36	3.0
German	3	3.25	0.55	0.94	32.08	4.83	0.82	1.4	**47.78**	4.38	0.74	1.27	43.34	10.1	1.71	2.93
	7	2.02	0.8	1.14	32.39	3.03	1.2	1.72	**48.86**	2.89	1.14	1.64	46.59	6.2	2.45	3.52
	10	1.58	0.89	1.14	31.93	2.39	1.35	1.73	48.46	2.42	1.37	1.75	**49.02**	4.95	2.8	3.57
French	3	2.48	0.43	0.74	27.82	4.86	0.85	1.45	54.51	5.04	0.88	1.5	**56.39**	8.93	1.56	2.66
	7	1.59	0.65	0.92	27.06	3.45	1.41	2.0	58.82	3.71	1.51	2.15	**63.24**	5.87	2.39	3.4
	10	1.3	0.76	0.96	27.75	3.13	1.82	2.3	66.47	3.18	1.85	2.34	**67.63**	4.69	2.73	3.46
Spanish	3	3.0	0.51	0.88	41.31	4.7	0.8	1.37	64.32	6.08	1.04	1.77	**83.1**	7.3	1.25	2.13
	7	1.98	0.79	1.13	37.54	3.61	1.44	2.06	68.44	4.45	1.77	2.54	**84.39**	5.28	2.11	3.01
	10	1.75	1.0	1.27	39.56	3.28	1.87	2.38	74.14	3.63	2.06	2.63	**81.93**	4.43	2.52	3.21
Arabic	3	3.09	0.51	0.87	68.5	3.35	0.55	0.95	74.8	5.28	0.87	1.49	**117.32**	4.5	0.74	1.27
	7	2.32	0.89	1.29	75.44	2.43	0.93	1.35	78.95	3.92	1.5	2.17	**126.9**	3.09	1.19	1.71
	10	1.85	1.02	1.31	72.38	2.05	1.12	1.45	80.11	3.01	1.65	2.13	**117.68**	2.55	1.4	1.81
Russian	3	1.37	0.23	0.39	43.33	1.65	0.27	0.47	52.22	1.78	0.3	0.51	**56.67**	3.16	0.53	0.9
	7	0.76	0.3	0.43	42.16	1.29	0.5	0.72	70.59	1.59	0.62	0.89	**87.25**	1.82	0.71	1.02
	10	0.7	0.39	0.5	44.64	1.11	0.62	0.79	70.54	1.56	0.87	1.12	**100.0**	1.56	0.87	1.12
Finance																
Italian	3	1.23	0.22	0.37	22.42	2.12	0.38	0.64	38.79	2.65	0.47	0.8	**48.48**	5.47	0.97	1.65
	7	0.83	0.34	0.49	17.31	1.81	0.75	1.06	37.46	2.04	0.84	1.19	**42.05**	4.84	2.0	2.83
	10	0.74	0.44	0.55	16.87	1.64	0.97	1.22	37.42	1.69	1.0	1.26	**38.65**	4.39	2.59	3.26
German	3	4.05	0.78	1.3	41.27	4.31	0.83	1.39	44.13	4.84	0.93	1.55	**49.21**	9.8	1.88	3.15
	7	2.24	1.0	1.38	35.94	2.63	1.18	1.63	42.45	3.31	1.48	2.04	**53.12**	6.22	2.78	3.84
	10	1.88	1.2	1.47	37.6	2.2	1.4	1.71	43.73	2.75	1.75	2.14	**54.73**	5.02	3.2	3.91
French	3	1.83	0.34	0.58	27.36	3.55	0.66	1.11	52.36	4.24	0.79	1.33	**62.74**	6.76	1.26	2.12
	7	1.18	0.51	0.71	22.98	2.65	1.15	1.61	52.1	3.73	1.62	2.26	**73.14**	5.11	2.22	3.09
	10	0.96	0.6	0.74	22.49	2.27	1.41	1.74	52.89	2.99	1.85	2.29	**69.6**	4.3	2.66	3.29
Spanish	3	2.11	0.37	0.63	24.14	3.56	0.63	1.07	41.0	4.87	0.86	1.46	**55.94**	8.7	1.54	2.61
	7	1.36	0.56	0.79	23.3	2.03	0.84	1.19	35.1	4.29	1.77	2.5	**73.75**	5.82	2.4	3.39
	10	1.15	0.67	0.85	23.22	1.9	1.12	1.41	38.52	3.56	2.09	2.64	**72.13**	4.94	2.91	3.66
Arabic	3	4.26	0.7	1.2	78.43	3.29	0.54	0.93	60.78	6.4	1.05	1.81	**118.3**	5.43	0.89	1.53
	7	2.82	1.08	1.57	89.71	2.74	1.05	1.52	86.86	3.99	1.53	2.21	**126.29**	3.16	1.21	1.75
	10	2.33	1.28	1.65	80.1	2.21	1.21	1.56	75.73	3.31	1.82	2.35	**114.08**	2.91	1.59	2.06
Russian	3	0.7	0.12	0.2	17.24	1.67	0.28	0.48	41.38	2.37	0.4	0.68	**58.62**	4.04	0.68	1.16
	7	0.96	0.37	0.54	36.49	1.43	0.56	0.8	54.05	1.85	0.72	1.04	**70.27**	2.63	1.03	1.48
	10	0.75	0.42	0.54	36.73	1.34	0.75	0.96	65.31	1.42	0.79	1.02	**69.39**	2.05	1.14	1.47

and Ground Truth separately for each domain. To deepen the analyses, Table 4 reports the Precision, Recall, and F1-measure scores for three representative K values (i.e., 3, 7, and 10) separately for each domain. Columns labeled as $\Delta F1vs.G.T.$ in Table 4 indicate, for each method, the percentage value ratio of the achieved F1-measure to the G.T. score. In most cases, both linear and non-linear methods outperformed the general-purpose model. The gap is particularly significant for specific European languages (e.g., French, Spanish), where the syntactic and semantic language similarities with the source language (English) provide clear benefits. Surprisingly, convincing results were achieved for non-European languages as well for all the analyzed domains (e.g., in Russian NLIN achieved 86% of the G.T. score for Medicine). This supports the hypothesis that word shifts due to domain adaptation are, to a large extent, predictable independently of language grammar and syntax. As expected, the non-linear model has shown to achieve better performance than the linear one in almost all languages and domains due to the inherent complexity of the inference task. In the Arabic language the Ground Truth performed slightly worse than the inference model according to the extrinsic evaluation scores. This is probably due to the higher morphological richness and to the increasing lexical ambiguity of the Arabic language compared to English, which have already been highlighted by previous studies related to Arabic Wikipedia content (e.g., [49]). The latter findings reinforce the need for alternative, algorithmic solutions to automatically infer domain-specific models, such as the newly proposed approach described by the present study.

E. PARAMETER ANALYSIS

We investigated the use of fully connected neural networks with different characteristics to tackle both the vector identification and inference problems.

Figure 5 plots the F1-measure scores achieved by the 2-layer deep neural network architectures characterized with different width (W) for the technology domain (chosen as representative). The results show that, independently of the

considered language, the performance is weakly influenced by the number of nodes per layer provided that it is above the number of inputs (300). Therefore, to limit the computational complexity of model training, hereafter we will set W to 900 (3 times the number of inputs) for all the considered languages.

Figure 6 shows the impact of the network depth, where we varied the number of hidden layers from 1 to 5. The best average performance was achieved by the 2- and 3-layer networks on most of the tested languages and domains. Typically, the level of complexity of the inference process seems to not require the use of more than 2 or 3 layers. For example, for the Arabic language the 4-layer Deep Learning architecture performed worst (see Figure 6c). Hence, to avoid data overfitting and to limit the computational time we recommend to use, as default setting, a 2-layer fully connected network.

VI. CONCLUSION AND FUTURE WORK

The paper proposed to infer aligned domain-specific Word2Vec embeddings in a multilingual scenario where, for *some* of considered languages, there is a lack a domain-specific data and/or pre-trained word vectors. Since, typically, this is *not* an issue for *all* languages but only for a subset of them, we proposed to opportunistically reuse the information provided by a source, data-rich language (e.g., English) to infer how word vectors should change in order to tailor general-purpose models to specific domains. An extrinsic evaluation carried out on a newly proposed benchmark dataset show that the proposed approach is able to effectively support word retrieval in a multilingual context.

The main takeaways from the experiments are enumerated below:

- Both Linear and Non-Linear models outperformed the General-Purpose one. While coping with document corpora relative to domains and languages for which the standard domain adaptation pipeline is not applicable, they bring clear benefits to the NLP process.
- For specific combinations of language and domain (e.g., French-Medicine, Russian-Technology), the best performing version of the proposed approach achieved results comparable to the Ground Truth. In few exceptional cases relative to the Arabic language, the inference-based model even beat the Ground Truth, probably due to the inherent complexity of the domain adaptation step.
- Non-Linear 2-layer fully connected deep models have shown to averagely perform best. They were able to capture non-linear word vector relationships without incurring in data overfitting.

The achieved results leave room for further improvements. Firstly, since multilingual data are often changing, we aim at studying how multilingual domain-specific word embeddings evolve over time account [34]. Secondly, we plan to apply the proposed methodology to address various cross-lingual NLP task among which cross-lingual text summarization and sentiment analysis, search engines, cross-lingual media retrieval, and conversational agents. Finally, we aim at leveraging the proposed inference-based approach to map the vector representations of multimodal content (e.g., videos, images).

ACKNOWLEDGMENT

Computational resources were provided by HPC@POLITO, a project of Academic Computing within the Department of Control and Computer Engineering at the Politecnico di Torino (http://www.hpc.polito.it).

REFERENCES

[1] D. Bollegala, T. Mu, and J. Y. Goulermas, "Cross-domain sentiment classification using sentiment sensitive embeddings," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 2, pp. 398–410, Feb. 2016.
[2] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer, "Multilingual denoising pre-training for neural machine translation," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 726–742, Dec. 2020.
[3] A. I. Kadhim, "Survey on supervised machine learning techniques for automatic text classification," *Artif. Intell. Rev.*, vol. 52, no. 1, pp. 273–292, Jun. 2019.
[4] H. Fei, S. Tan, and P. Li, "Hierarchical multi-task word embedding learning for synonym prediction," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Jul. 2019, pp. 834–842.
[5] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[6] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.
[7] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[8] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 5754–5764.
[9] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2018, pp. 2227–2237.
[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805
[11] G. Kruszewski, I.-T. Sorodoc, and T. Mikolov, "Evaluating online continual learning with CALM," 2020, *arXiv:2004.03340*. [Online]. Available: http://arxiv.org/abs/2004.03340
[12] F. Nooralahzadeh, L. Øvrelid, and J. T. Lønning, "Evaluation of domain-specific word embeddings using knowledge resources," in *Proc. 11th Int. Conf. Lang. Resour. Eval. (LREC)*, 2018, pp. 1–8.
[13] Y. Yang and J. Eisenstein, "Unsupervised multi-domain adaptation with feature embeddings," in *Proc. Conf. North Amer. Assoc. Comput. Linguistics, Hum. Language Technol.*, Denver, CO, USA, R. Mihalcea, J. Y. Chai, and A. Sarkar, Eds., May/Jun. 2015, pp. 672–682.
[14] E. Khabiri, W. M. Gifford, B. Vinzamuri, D. Patel, and P. Mazzoleni, "Industry specific word embedding and its application in log classification," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, Nov. 2019, pp. 2713–2721.
[15] X. Liu, J.-Y. Nie, and A. Sordoni, "Constraining word embeddings by prior knowledge–application to medical information retrieval," in *Proc. Asia Inf. Retr. Symp.* New York, NY, USA: Springer, 2016, pp. 155–167.
[16] S. J. Pan, Z. Toh, and J. Su, "Transfer joint embedding for cross-domain named entity recognition," *ACM Trans. Inf. Syst.*, vol. 31, no. 2, pp. 1–27, May 2013.
[17] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," in *Proc. 11th Int. Conf. Lang. Resour. Eval.*, Miyazaki, Japan, N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, Eds., May 2018, pp. 1–5.

[18] R. Al-Rfou', B. Perozzi, and S. Skiena, "Polyglot: Distributed word representations for multilingual NLP," in *Proc. 17th Conf. Comput. Natural Lang. Learn.*, Sofia, Bulgaria, J. Hockenmaier and S. Riedel, Eds., Aug. 2013, pp. 183–192.

[19] A. Joulin, P. Bojanowski, T. Mikolov, H. Jégou, and E. Grave, "Loss in translation: Learning bilingual word mapping with a retrieval criterion," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 2979–2984.

[20] X. Dong, Y. Zhu, Y. Zhang, Z. Fu, D. Xu, S. Yang, and G. de Melo, "Leveraging adversarial training in self-learning for cross-lingual text classification," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, New York, NY, USA, Jul. 2020, pp. 1541–1544.

[21] A. Ramachandran and G. de Melo, "Cross-lingual emotion lexicon induction using representation alignment in low-resource settings," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 5879–5890.

[22] Y. Cao, H. Liu, and X. Wan, "Jointly learning to align and summarize for neural cross-lingual summarization," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds., Jul. 2020, pp. 6220–6231.

[23] D. D. Lewis and K. S. Jones, "Natural language processing for information retrieval," *Commun. ACM*, vol. 39, no. 1, pp. 92–101, 1996.

[24] N. Schluter, "The word analogy testing caveat," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 2, New Orleans, LA, USA, M. A. Walker, H. Ji, and A. Stent, Eds., Jun. 2018, pp. 242–246.

[25] C. Allen and T. M. Hospedales, "Analogies explained: Towards understanding word embeddings," in *Proc. 36th Int. Conf. Mach. Learn.*, vol. 97, Long Beach, CA, USA, K. Chaudhuri and R. Salakhutdinov, Eds., Jun. 2019, pp. 223–231.

[26] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6338–6347.

[27] Z. S. Harris, "Distributional structure," *Word*, vol. 10, nos. 2–3, pp. 146–162, 1954.

[28] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Feb. 2003.

[29] J. Li, J. Li, X. Fu, M. A. Masud, and J. Z. Huang, "Learning distributed word representation with multi-contextual mixed embedding," *Knowl.-Based Syst.*, vol. 106, pp. 220–230, Aug. 2016.

[30] W. L. Hamilton, K. Clark, J. Leskovec, and D. Jurafsky, "Inducing domain-specific sentiment lexicons from unlabeled corpora," in *Proc. Conf. Empirical Methods Natural Lang. Process.* 2016, pp. 595–605.

[31] S. Li, R. Pan, H. Luo, X. Liu, and G. Zhao, "Adaptive cross-contextual word embedding for word polysemy with unsupervised topic modeling," *Knowl.-Based Syst.*, vol. 218, Apr. 2021, Art. no. 106827.

[32] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Lifelong domain word embedding via meta-learning," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 4510–4516.

[33] D. Bollegala, T. Maehara, and K.-I. Kawarabayashi, "Unsupervised cross-domain word representation learning," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics*, Beijing, China, 2015, pp. 730–740.

[34] K. Jha, G. Xun, V. Gopalakrishnan, and A. Zhang, "DWE-med: Dynamic word embeddings for medical domain," *ACM Trans. Knowl. Discovery Data*, vol. 13, no. 2, pp. 1–21, Jun. 2019.

[35] K. Komiya, S. Suzuki, M. Sasaki, H. Shinnou, and M. Okumura, "Domain adaptation for word sense disambiguation using word embeddings," in *Computational Linguistics and Intelligent Text Processing*, A. Gelbukh, Ed. Cham, Switzerland: Springer, 2018, pp. 195–206.

[36] R. Dabre, C. Chu, and A. Kunchukuttan, "A survey of multilingual neural machine translation," *ACM Comput. Surv.*, vol. 53, no. 5, pp. 99:1–99:38, 2020.

[37] E. Grave, A. Joulin, and Q. Berthet, "Unsupervised alignment of embeddings with wasserstein procrustes," in *Proc. 22nd Int. Conf. Artif. Intell. Statistics*, 2019, pp. 1880–1890.

[38] D. Alvarez-Melis and T. Jaakkola, "Gromov-wasserstein alignment of word embedding spaces," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1881–1890.

[39] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou, "Word translation without parallel data," 2017, *arXiv:1710.04087*. [Online]. Available: http://arxiv.org/abs/1710.04087

[40] J. Goikoetxea, A. Soroa, and E. Agirre, "Bilingual embeddings with random walks over multilingual wordnets," *Knowl.-Based Syst.*, vol. 150, pp. 218–230, Jun. 2018.

[41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Jun. 2019, pp. 4171–4186.

[42] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *Proc. 5th Int. Conf. Learn. Represent.*, Toulon, France, Apr. 2017, pp. 1–16.

[43] L. Zhang, S. Zhang, and K. Balog, "Table2 Vec: Neural word and entity embeddings for table population and retrieval," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, New York, NY, USA, Jul. 2019, pp. 1029–1032.

[44] L. Cagliero, A. Fiori, and L. Grimaudo, "Personalized tag recommendation based on generalized rules," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 1, pp. 12:1–12:22, 2013.

[45] W. Yu, X. Lin, J. Ge, W. Ou, and Z. Qin, "Semi-supervised collaborative filtering by text-enhanced domain adaptation," in *Proc. 26th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Virtual Event, CA, USA, R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, Eds., Aug. 2020, pp. 2136–2144.

[46] M. J. Zaki and W. Meira, *Data Mining Machine Learning: Fundamental Concepts Algorithms*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2020.

[47] R. Řehůřek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proc. LREC Workshop New Challenges NLP Frameworks*, Valletta, Malta, May 2010, pp. 45–50.

[48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[49] B. Mohit, N. Schneider, R. Bhowmick, K. Oflazer, and N. A. Smith, "Recall-oriented learning of named entities in Arabic Wikipedia," in *Proc. 13th Conf. Eur. Chapter Assoc. Comput. Linguistics*, New York, NY, USA, 2012, pp. 162–173.

LUCA CAGLIERO (Member, IEEE) received the master's degree in computer and communication network and the Ph.D. degree in computer engineering both from the Politecnico di Torino. Since January 2020, he has been an Associate Professor with the Dipartimento di Automatica e Informatica, Politecnico di Torino. He teaches B.Sc., master-level, and Ph.D. courses on database systems and data mining techniques. Specifically, he has worked on text summarization, classification, and association rule mining. He has coauthored more than 100 scientific articles, including more than 40 international journals. His current research interests include machine learning, text mining, and deep NLP. He has been the contact person of various consulting and research contracts. He was a recipient of the Working Capital Research Grant 2012 for the research project on Web document summarization. He is currently an Associate Editor of the *ESWA* (Elsevier) and *MLWA* (Elsevier) journals.

MORENO LA QUATRA (Member, IEEE) is currently pursuing the Ph.D. degree with the Politecnico di Torino. After his graduation in computer engineering with a double degree program between Politecnico di Torino and Grenoble INP, he started his Ph.D. in the domain of multimedia and text analysis. His main research interests include deep neural networks for language modeling and understanding. He has served as a Reviewer for the *ESWA* (Elsevier), *KBS* (Elsevier), *FGCS* (Elsevier), IEEE Access, and the *KAIS* (Springer) journals.