Journal of Intelligent Information Systems
https://doi.org/10.1007/s10844-023-00814-z

RESEARCH

OIE4PA: open information extraction for the public administration

Lucia Siciliani[1] · Eleonora Ghizzota[1] · Pierpaolo Basile[1] · Pasquale Lops[1]

Received: 22 May 2023 / Revised: 21 August 2023 / Accepted: 22 August 2023

Abstract
Tenders are powerful means of investment of public funds and represent a strategic development resource. Despite the efforts made so far by governments at national and international levels to digitalise documents related to the Public Administration sector, most of the information is still available in an unstructured format only. With the aim of bridging this gap, we present OIE4PA, our latest study on extracting and classifying relations from tenders of the Public Administration. Our work focuses on the Italian language, where the availability of linguistic resources to perform Natural Language Processing tasks is considerably limited. Nevertheless, OIE4PA adopts a multilingual approach so it can be applied to several languages by providing appropriate training data. Rather than purely training a classifier on a portion of the extracted relations, the backbone idea of our learning strategy is to put a supervised method based on self-training to the proof and to assess whether or not it improves the performance of the classifier. For evaluation purposes, we built a dataset composed of 2,000 triples which have been manually annotated by two human experts. The in-vitro evaluation shows that OIE4PA achieves a MacroF$_1$ equal to **0.89** and a **91**% accuracy. In addition, OIE4PA was used as the pillar of a prototype search engine, which has been evaluated through an in-vivo experiment with positive feedback from 32 final users, obtaining a SUS score equal to **83.98**.

Keywords Open information extraction · Natural language processing · Information retrieval · Public administrations · Self-training

✉ Lucia Siciliani
lucia.siciliani@uniba.it

Eleonora Ghizzota
e.ghizzota@studenti.uniba.it

Pierpaolo Basile
pierpaolo.basile@uniba.it

Pasquale Lops
pasquale.lops@uniba.it

[1] Department of Computer Science, University of Bari Aldo Moro, via Edoardo Orabona, Bari 70125, Bari, Italy

Published online: 20 September 2023

1 Introduction

The beneficial impact of Artificial Intelligence (AI) within the public sector domain is well-known in the literature (Dwivedi et al., 2021; Kalampokis et al., 2023). In particular, Machine learning (ML) and Natural Language Processing (NLP) are the methodologies that are most used to support the public sector (Madan & Ashok, 2023). However, despite the efforts already made to digitalise and exploit the information currently used by Public Administrations (PAs), much work is still needed to achieve a satisfying result. In one of the latest studies conducted by *Interoperable Europe*[1], an initiative of the European Commission for a reinforced public sector interoperability policy concerning the current use of NLP solutions in Public Services (Barthélemy et al., 2022), the authors have identified two main challenges: the lack of a unified ontology to properly represent the domain in applications and the under-representation of several European languages (English, Spanish, and French being the only ones for which many resources are available).

With the aim of bridging both of these gaps, we present OIE4PA (Open Information Extraction for Public Administrations), a framework whose underlying purpose is to extract structured facts from announcements of the Public Administration. OIE4PA adopts the methodology shown in WikiOIE (Siciliani et al., 2021) and extends it to cover the domain of public tenders. Open Information Extraction (Niklaus et al., 2018) (Open IE) is the task of extracting facts from unstructured texts by generating a machine-readable representation of the information contained within them. A fact is usually in the form of a triple or n-ary proposition and defines a relation between two entities. Open IE is essential in several Natural Language Processing applications and represents a first step towards the automatic creation of ontologies from text.

Differently from traditional Relation Extraction tasks, in which the relations to extract are known in advance, Open IE systems like OIE4PA can extract any relation in the corpus without prior input information. This feature is fundamental nowadays, where the amount of textual data available online is growing at an outstanding rate. Moreover, considering specific domains like Public Administration, the adopted vocabulary is highly specific, thus making it infeasible to compile a list of all the relations of interest.

To address this issue, we set up specific tools and extracted facts from unstructured sources, i.e., textual documents and scans of tenders acquired from the EmPulia[2] platform. A portion of the triples extracted from such documents was annotated by human experts to build training and test sets. We investigate several machine learning techniques for classifying relevant and non-relevant extracted triples by obtaining considerable results in terms of accuracy. In addition, OIE4PA was integrated within a search engine prototype, allowing us to perform an in-vivo evaluation with our expected final users.

1.1 Research objective and contributions

The main research objective of this work is to bridge the aforementioned gaps in the availability of solutions when dealing with restricted domains like the public sector and languages that differ from English; furthermore, we attempt to explore the impact of an Open IE system in a domain such as the procurement sector, where documents can consist of many pages and the writing style might be wordy and complex due to bureaucratic vocabulary.

[1] joinup.ec.europa.eu/interoperable-europe

[2] www.empulia.it

Thus, the principal contributions of this paper are the following:

- We adopt an Open IE system that allows extraction and classification of relations from the public tenders domain. OIE4PA adopts an unsupervised approach based on linguistic features for extracting relations and a supervised approach combined with self-training for classifying relevant triples. In particular, we deeply investigate several machine learning approaches to identify relevant and non-relevant triples in the public tenders domain extracted by the unsupervised Open IE system;
- We explore the potentialities of self-training and make a comparison with the typical supervised approach in terms of performance. As a matter of fact, manually annotating data can be time-consuming and expensive. To overcome these complications, a semi-supervised approach, e.g. self-training, can be employed, so that unlabelled data can still be exploited for the training task. OIE4PA obtains encouraging results when trained on automatically annotated triples.
- We make publicly available a dataset of 2,000 labelled triples extracted from Italian announcement documents. The quality of the labels is high since the annotation was performed by two human Italian native speakers;
- We developed a search engine encapsulating our Open IE model with the purpose of helping domain experts in their work activities. Therefore the engine has been evaluated through an in-vivo study involving 32 participants, obtaining very positive feedback.

The paper is structured as follows: Section 2 offers an overview of the state of the art regarding Open IE systems for English and Italian, Section 3 deals with the details of the OIE4PA pipeline. In Section 4 we show and discuss the results obtained by OIE4PA in both an in-vitro and in-vivo evaluation. Finally, Section 5 and Section 6 conclude this work and illustrates our future research directions.

2 Related work

In the early stages of Information Extraction, the main task was to automatically extract structured information from unstructured and semi-structured machine-readable sources and to represent it in a tuple of two *entities* and a *relation* between them, namely *relation(entity1, entity2)*. It is essential to point out that IE systems focus on predefined guidelines specifying which objects and relations in a homogeneous source are relevant to the application. Since IE systems extract tuples from domain-specific corpora, their performance considerably depends on domain knowledge, as relations have to be specified as their input. Traditional IE systems rely indeed on linguistic technologies tuned to the domain of interest (Banko et al., 2007), hence shifting to a different domain implies starting afresh. While this human-involving task is feasible for narrow domains, enumerating every potential relation is much more intractable in large and varied corpora available on the Web, which contain a greater number of complex entity types. This limit prevents IE from extracting relation tuples across different domains without prior knowledge. It follows that scalability and portability across domains are not IE systems' main strengths. First-generation Open IE techniques (Vo & Bagheri, 2016) were introduced to tackle these drawbacks. The main goal was to develop a highly scalable system on large Web corpora, constructing a general, relation-independent model able to express a relation based on unlexicalised features, e.g., Part-of-Speech, shallow tags, surrounding context of a verb, capitalisation, and punctuation. This new paradigm only requires the corpus as input and no predefined set of relations. These features allow targeting *all relations* without further human input.

Examples of first-generation Open IE systems are TextRunner (Banko et al., 2007), WOE (Wu & Weld, 2010), which uses Wikipedia as a source of training data, StatSnowBall (Zhu et al., 2009), and SRLIE (Christensen et al., 2010), based on semantic role labelling. At this stage, Open IE can operate without knowing the focus relations *a priori* and can extract all relations simultaneously. Despite this progress, the first-generation Open IE systems still need some refinements; in fact, they do not always successfully extract the full relation between two noun phrases, or they only identify a portion of the relation, making it ambiguous.

Second-generation Open IE systems aim to fine-tune prior paradigms in order to overcome their aforementioned limits: incoherent extractions and uninformative extractions (Etzioni et al., 2011; Vo & Bagheri, 2016). What differentiates the first and second-generation Open IE systems is that the latter focus deeply on a thorough linguistic analysis of sentences, obtaining significantly higher performance. Several systems have been proposed that typically follow two main extraction paradigms: ReVerb and OLLIE (Mausam et al., 2012) implement a verb-based relation extraction (Christensen et al., 2011), while ClausIE (Corro & Gemulla, 2013) adopts a clause-based relation extraction solution. In order to limit error propagation and make extractable relations more flexible, Cabot and Navigli (2021) presented REBEL, an autoregressive seq2seq model which extracts triples as a sequence of text by performing an end-to-end relation extraction. A similar generative approach proposed by Josifoski et al. (2021) introduces an end-to-end autoregressive model for closed information extraction. On the other hand, a combination of generative approaches with reinforcement learning is described in Dognin et al. (2021). Despite the good performance of generative approaches, they work well when the subject, predicate, and object are structured sub-strings aligned with a KB (Gashteovski et al., 2020). In fact, generative approaches rely on large knowledge graphs, e.g., Wikidata or DBpedia, to automatically build training data.

Open IE systems listed so far have been developed or tested specifically for English. Expanding the range of supported languages means adapting an Open IE system to different grammatical features, syntax, lexica, and knowledge graphs. Since this paper is focused on Open IE for Italian, we now focus on the solutions proposed to address this issue. Although Italian is a major European language, no significant research has been conducted on Italian Open IE until the last decade. In 2018, Damiano et al. (2018) introduced ItalIE, an Italian Open IE system whose aim is to extract n-ary coherent propositions from simple sentences made up of single clauses, which are classified following seven patterns defined compliantly with the features of the Italian language, e.g., Italian clauses admit the absence of the subject, unlike English clauses. Depending on these patterns, the corresponding minimal clause types are identified and used to guide the extraction of minimal clauses. At last, one or more propositions are generated from the input sentences with the addition of complements and adverbials. ItalIE does not require training data and allows sentences to be processed automatically in parallel. The extraction paradigm of the proposed system is deeply inspired by ClauseIE, which adopts a clause-based paradigm. ClausIE uses clause structures to extract relations and their arguments, as well as dependency parsers and a set of rules for domain-independent lexica to locate clauses in sentences and separate the information in coherent propositions.

A few years after ItalIE, Guarasci et al. (2020) proposed a new linguistic-based, unsupervised system designed to identify elementary tuples and all their permutations from elementary one-verb sentences written in Italian, maintaining the grammaticality and the acceptability. The proposed approach uses complex linguistic structures and dependency parsers to detect verbal behaviour patterns; moreover, it is based on the derived patterns and is arranged in a pipeline consisting of four steps, sentence processing, argument identification, pattern recognition, and proposition generation.

3 Methodology

Open Information Extraction can be seen as the first step that allows the transformation of unstructured data into its structured form. In fact, its main goal is to extract facts from texts which usually take the form of triples composed of a *subject*, a *predicate*, and an *object*.

As stated in Section 1, Public Administration is a sector where the digitalisation level is still very low despite the efforts made until now, especially in Italy. Therefore, this kind of domain can benefit the most from applying NLP techniques.

Figure 1 shows an example of triples extracted from a Public Procurement attachment downloaded from the EmPulia website. An example of one of these triples is "i parapetti devono avere un'altezza di 1,00m" (*railings must have a height of 1,00 meter*) where the subject is "i parapetti" (*railings*), the predicate is "devono avere" (*must have*), and the object is "un'altezza di 1.00m" (*a height of 1,00 meter*). Documents related to a PA procedure can consist of several pages, e.g., the document used for the previous example is composed of 1,039 pages, and there is a lack of a unified template even among the same typology of documents. It is clear that, under these circumstances, also domain experts, i.e., people working daily in the PA domain, can have several issues when trying to find a specific piece of information.

On the grounds of this context, Open IE can surely help to retrieve relevant information more quickly. Triples extracted from tenders and other PA documents can also be used for various applications, ranging from visualisation and knowledge base completion to more advanced inference techniques. In this scenario, we mainly focus on the retrieval aspect and the support this research can offer PA professionals. Figure 2 shows an overview of the

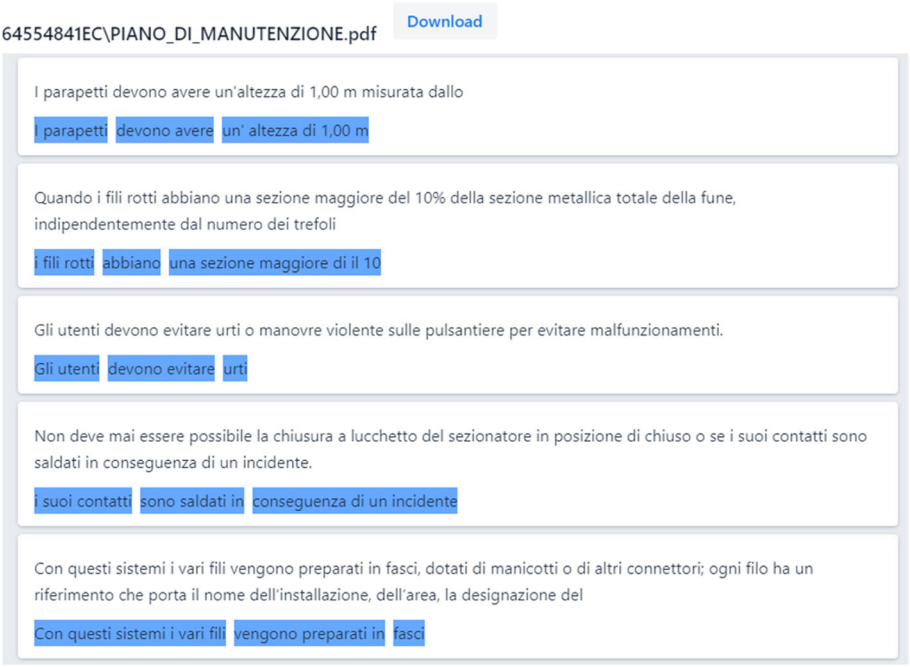

Fig. 1 List of the triples extracted from one document

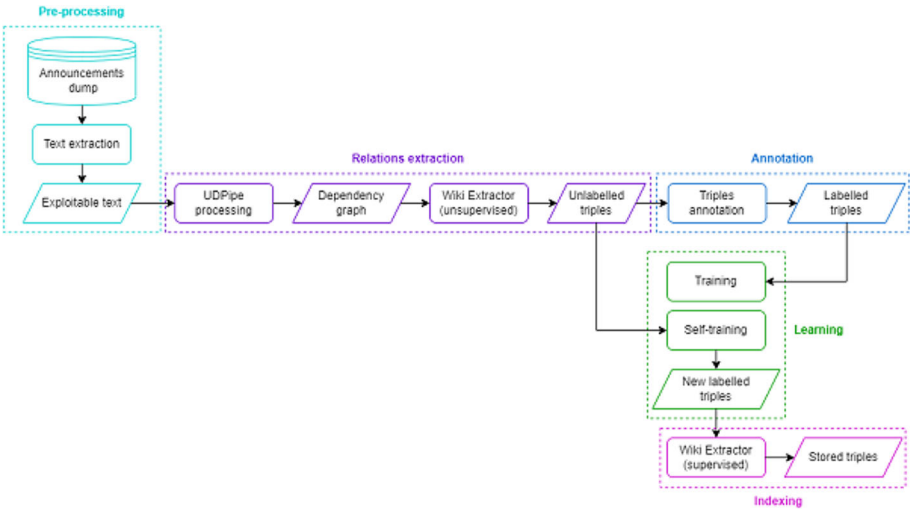

Fig. 2 OIE4PA pipeline

OIE4PA pipeline. OIE4PA adapts the methodology used in WikiOIE (Siciliani et al., 2021) to the Public Administration domain. The proposed system primarily relies on a supervised classification approach augmented by a self-training strategy. One of the main strengths of this approach is that even though our study focuses on the Italian language, it can be applied to several languages by providing appropriate training data.

3.1 Pre-processing

First and foremost, plain text has been extracted from a collection of PA announcement documents to extract relations. However, even before starting the text extraction, some precautions had to be taken for the system to work correctly. Many input files were compressed into archives or encrypted in .p7m format. The library used for text extraction, Apache Tika[3], does not accept the type of files mentioned above as input. In order to not lose these resources and elaborate them, we developed a script that decompresses the archives and then decrypts the .p7m files. Thereafter, although the documents were made directly accessible, many of them contained scans of the actual announcements. Apache Tika takes this issue into account and, through Tesseract OCR[4], can convert scans of typed text into machine-encoded text. In particular, OIE4PA supports text extraction from .doc, .docx, .ppt, .pptx, .xls, .xlsx, .pdf, .jpg, .png and .txt formats.

3.2 Relation extraction

The extracted plain text is read line-by-line and processed by the UDPipe tool (Straka & Straková, 2017). Each line corresponds to a paragraph identified by Tika. UDPipe is a trainable pipeline for sentence detection, tokenisation, tagging, lemmatisation, and dependency

[3] tika.apache.org/

[4] tesseract-ocr.github.io/

Table 1 Patterns of valid predicates

PoS-tag pattern	Example
AUX VERB ADP	*... è nato nel... (was born in)*
AUX VERB	*... è nato... (was born)*
AUX = (to be)	*... è... (is)*
VERB ADP	*... pensò a... (thought of)*
VERB	*... scrisse... (wrote)*

(AUX = auxiliary, VERB = verb, ADP = apposition)

parsing of CoNLL-U (Buchholz & Marsi, 2006) files. Universal Dependencies[5] can be used for training UDPipe: this allows us to obtain PoS-tags and syntactic dependencies that are annotated with a shared set of tags belonging to several languages.

The output of each sentence is used to generate a dependency graph for the module that extracts facts in the form of (*subject, predicate, object*). Triples are obtained with an unsupervised strategy based on PoS-tag and syntactic dependencies. More details about this approach are described in our previous work (Cassotti et al., 2021). First, the system identifies a sequence of PoS-tags that matches one of the verb patterns listed in Table 1. For example, "Il concorrente **dovrà presentare** proposte migliorative" (*The competitor* **has to present** *meliorative offers*) matches the **AUX VERB** pattern, so the algorithm identifies it as the predicate.

After a successful predicate match, the system checks for its respective subject and object; these must be compliant with the following constraints:

- there must be a sequence of specific PoS-tag tokens, i.e., noun, adjective, number, (only one) determiner, (only one) apposition, proper noun;
- the candidate subject must precede the predicate;
- the candidate object must follow the predicate.

Looking at the previous example, "**Il concorrente**" (*the competitor*) is identified as the subject, while "**proposte migliorative**" (*meliorative offers*) is the object, since they both respect the above-mentioned constraints. When the appropriate subject and object have been identified, two validation strategies can be applied:

- **simple**: subject, object, and predicate are accepted as a valid triple, and a score is assigned to it. To compute this score, two separate scores are calculated for the subject and the object:
  - **+1** for each *noun* occurring in the subject and the object;
  - **+2** for each *proper noun* occurring in the subject and the object;

  Proper nouns have a higher score since they can indicate the presence of a named entity which is the subject or the object of the triple. These values are then multiplied by $\frac{1}{l}$, where l is the number of tokens occurring in the subject and the object. Scores are not normalised to give a further boost to named entities composed of proper nouns. The final triple score is the sum of the subject and object scores. In this way, subjects and objects including only nouns and proper nouns are more relevant;
- **simpledep**: the triple is accepted only if there is a syntactic relation in the dependency tree between the predicate and both the subject and the object. The triple score is computed as illustrated before. It is possible that the syntactic relation is not *subj* or *obj*, for example

[5] universaldependencies.org

in the sentence "*Il pagamento del corrispettivo contrattuale dovrà avvenire sul conto corrente*[6]." the triple object "sul conto corrente" is linked to the triple predicate "dovrà avvenire" by the *obl (oblique nominal)*[7] relation. Other cases involve the syntactic relation *xcomp (open clausal complement)*[8].

Our proposed system applies the **simpledep** validation method, whose results are stored in JSON format. The following listing shows an example of a triple extracted with simpledep.

Listing 1 Example of triple extracted with the simpledep validation method.

```
{"id":"3020",
    "title":"C:\...\Allegato_all.A_Disciplinare_di_gara_.pdf",
    "text":"Il concorrente rende tutte le informazioni richieste mediante la
    compilazione delle parti pertinenti.",
    "triples":[
        {"subject":{"span":"Il concorrente","start":0,"end":2,"score":1.0},
        "predicate":{"span":"rende","start":2,"end":3,"score":1.0},
        "object":{"span":"tutte le informazioni","start":3,"end":6,
        "score":0.6666667},
        "score":1.6666667}
    ]}
```

Using this unsupervised extraction method, we processed 6,262 announcement documents for a total of 5,693,839 sentences and 2,912,973 of them matched the predicate patterns. Finally, applying the **simpledep** approach, we obtained 98,079 unlabelled triples.

3.3 Annotation

A subset of 2,000 triples was randomly selected from the original set of unlabelled ones; only triples containing a predicate with at least 20 exact matches in the entire set have been considered. We apply this threshold to filter rare predicates and have enough examples of each predicate in our dataset. From now on, we will refer to it as the L dataset. The triples contained in L have been manually annotated as *relevant* or *non-relevant* by two Italian native-speaker experts while a third annotator resolved the conflicting annotations. The triples containing typing errors, such as missing accents, misspelled and truncated words, have been replaced so that they do not negatively affect the training procedure. A triple is considered relevant if it is compliant with the following guidelines:

- the triple is grammatically correct;
- the triple is semantically correct;
- the identified subject, predicate, and object do actually have the role of subject, predicate, and object in the original sentence.

Examples of relevant triples are (*Il concorrente, dovrà presentare, proposte migliorative*)[9], (*L'aggiudicatario, potrà disporre di, tali dati*)[10], and (*L'esecutore, ha, il diritto*)[11]: these tuples are compliant with the guidelines listed before. On the other hand, triples like (*Partita,*

[6] In English: *"Payment of the contractual consideration must be made to the current account."*

[7] Oblique nominal: https://universaldependencies.org/u/dep/obl.html

[8] https://universaldependencies.org/u/dep/xcomp.html

[9] (The competitor, has to present, meliorative offers)

[10] (The awardee, will have, such data)

[11] (The executor, has, the right)

Iva, 6391740724)[12], (*massimo 30, facciate, formato A4 eslcuse*)[13] are not grammatically correct since the words *Iva* and *facciate* are not verbs and do not represent a predicate. Another example of an invalid triple is (*Di seguito, sono riportati, gli altri valori*)[14] where the predicate is correctly extracted and is a proper predicate but the triple is not self-explanatory so it is not semantically valid.

To measure the agreement between the aforementioned two expert annotators, we employed Cohen's Kappa coefficient κ, achieving a value of 0.77. A value of κ between 0.61 and 0.80 indicates a substantial agreement (Landis & Koch, 1977) which proves a good annotations quality. The third annotator revised 182 triples (0.09% of the total) in which the two annotators disagreed. Out of all the 2,000 triples in L, 1,380 were labelled as relevant (69% of the total), while the remaining 620 triples (31%) were labelled as non-relevant.

Later on, L is split into a training set and a test set; notice that L does not contain duplicates. The training set takes the 80% of L's triples, while the other part makes up the test set; therefore, we obtain a training set of $1,600$ triples and a test set of 400 triples; furthermore, they maintain the same balance of relevant and non-relevant triples as L, so they both contain the 69% of relevant triples and the 31% of non-relevant triples.

Lastly, we generate another dataset, referred to as U, by selecting from the initial unlabelled dataset the 20% of triples in which the predicate occurs at least 10 times. U must not contain any of the triples already included in L, so we deleted every possible duplicate, obtaining a final dataset of 14,096 unlabelled triples.

3.4 Learning

The backbone of our learning strategy is to put a supervised method based on self-training to the proof and assess whether it improves the classifier's performance (Siciliani et al., 2021).

Manually annotating data can be time-consuming and expensive. To overcome these complications, a semi-supervised approach can be leveraged. In semi-supervised learning, a classifier is trained on a small amount of labelled data and is then used to make predictions on the unlabelled data. In this way, unlabelled data can be exploited to a greater extent as an augmented training set. Self-training is a specific semi-supervised technique. When operating with a self-training algorithm, a classifier is learned iteratively by attributing pseudo-labels with a prediction confidence score to a set of unlabelled samples; the pseudo-labelled examples with a score higher than a given threshold are used to enrich the labelled training set (Amini et al., 2022).

We carry out a self-training strategy by leveraging the training, test, and U datasets described in Section 3.3 and by making use of three distinct supervised classifiers, i.e., L2-regularised logistic regression, L2-regularised L2-loss support vector classification implemented by LibLinear[15] (Fan et al., 2008), and gradient boosting implemented by XGBoost[16] (Chen & Guestrin, 2016).

Our self-training pipeline is built as follows:

[12] (VAT, number, 6391740724)

[13] (maximum 30, pages, in A4 dimension excluded)

[14] (Hereafter, are displayed, the other values)

[15] www.csie.ntu.edu.tw/ cjlin/liblinear/

[16] xgboost.readthedocs.io/en/stable/index.html

1. for each of the 20 iterations, the system randomly selects $p = 200$ triples from U, and the classifier, previously trained on the manually labelled training set, makes a prediction of the target class and assigns a confidence score to each of them;
2. the triples with a confidence score higher or equal to a threshold t are added to the original training set, preserving the classes' balance.

At the end of the process, the augmented training set contains the manually [17]labelled triples and the triples labelled by the trained classifier. Given a triple $< S, P, O >$ composed of subject (S), predicate (P) and object (O), the features used to represent each training example are the following:

- the PoS-tags occurring into S, O, and P;
- the n-gram that composes P;
- the path of syntactic dependencies that links S to P;
- the path of syntactic dependencies that links O to P;
- the PoS-tags of the 3-gram of the sequence preceding S;
- the PoS-tags of the 3-gram of the sequence following O;
- the concatenation of three embeddings: for S, O, and P, a word embedding is built by averaging all the embeddings of the words occurring in S, O, and P, respectively. The embeddings are retrieved from the Italian model of fastText (Bojanowski et al., 2016).

Given the following triple (*la stazione appaltante, deve esplicitare, il percorso*) (the contracting authority, must clarify, the steps) and the sentence in which it occurs "*...obbligatoria e che la stazione appaltante deve esplicitare il percorso seguito per la...*" (...mandatory and that the contracting authority must clarify the steps followed for the...), we extract the following features:

- PoS-tags in the subject: DET, NOUN, ADJ;
- PoS-tags in the predicate: AUX VERB;
- PoS-tags in the object: DET, NOUN;
- n-gram of the predicate: *deve_esplicitare* (must_clarify);
- Syntactic dependencies between subject and predicate: *nsubj*;
- Syntactic dependencies between object and predicate: *obj*;
- PoS-tags of the 3-gram preceding the subject: "*obbligatoria e che*" (mandatory and that), ADJ_CCONJ_SCONJ;
- PoS-tags of the 3-gram following the object: "*seguito per la*" (followed for the), ADJ_ADP_DET;
- a vector of floats that is the concatenation of three embeddings.

To investigate more recent text classification approaches based on Transformers and Large Language Models (LLMs), we fine-tune an Italian BERT (Kenton & Toutanova, 2019) model on the relevant and non-relevant triples dataset. In particular, as input of the model, we provide the extracted triple by concatenating the subject, predicate and object. During the concatenation, we add some meta tokens to identify the start of each part of the triples. For example, given the following triple *(la stazione appaltante, deve esplicitare, il percorso)* the input triple that feed the model is $< S >$ *la stazione appaltante* $< P >$ *deve esplicitare* $< O >$ *il percorso*. Then the LLM is fine-tuned using the training data. More details about the training procedure is reported in Section 4.3.

[17] fasttext.cc/docs/en/crawl-vectors.html

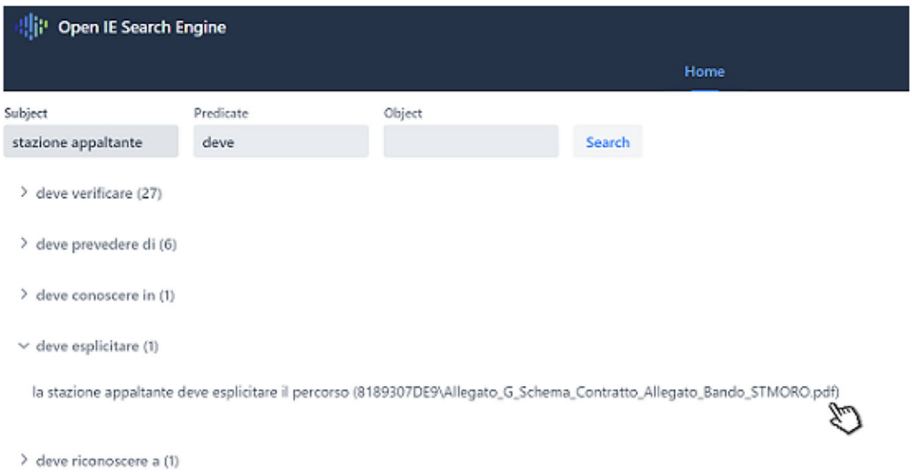

Fig. 3 The system shows the output triples grouped by their predicate. The user can then click on one of the predicates and visualise the entire sentence containing the triple

3.5 Indexing and search

In order to visualise the triples and allow users to search and browse them, we developed a search engine based on Lucene specialised in Italian PA tenders.[18] By filling in the fields **Subject**, **Predicate** and **Object**, either separately or jointly, it is possible to search for a specific word or sequence in the corpus of tenders documents.

Regardless of how many and which fields the user fills in, the search result is a list of predicates; the number in brackets indicates how many sentences match that predicate. As shown in Fig. 3, after selecting a predicate, the system shows the sequences containing the subject, predicate, and object specified beforehand.

Finally, when clicking on a particular sequence, the system displays the integral sentence, the sequence has been extracted from and highlights its subject, predicate, and object; furthermore, the user can visualise other sentences included in the document and download them (Fig. 4).

4 Evaluation

Results are evaluated in terms of *Precision* (P), *Recall* (R), F_1*-Measure* (F_1) for both non-relevant (0) and relevant (1) classes; *Macro* F_1 (MaF_1) and *Accuracy* (Acc.) for a comprehensive overview. The evaluation pipeline follows the ensuing procedure separately for each supervised learning method mentioned above. The classifier, the optimal parameters, and the test set are the same for all the phases of training, self-training, and testing. This simplifies the pipeline and avoids redundant information during each evaluation. Last but not least, our criterion for selecting the optimal value combinations is to settle on the ones resulting in the higher $MacroF_1$ score. Given these premises, the evaluation follows these steps:

[18] 193.204.187.101:8080/wikiopeniesearchengine-1.0-SNAPSHOT/

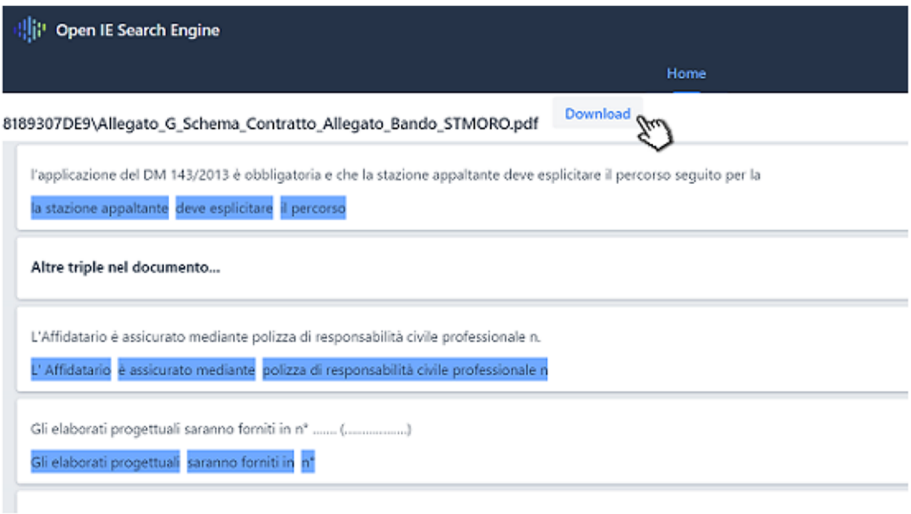

Fig. 4 The system visualises the sentence to which the selected triple belongs and all the other triples appearing in the same document. The user can also download the original file from which the triples were extracted

1. we train and test with 5-fold cross-validation the learning method using a set of potential optimal values for each parameter to be tuned, then we select the optimal combination;
2. during self-training, a further classification model is trained on the same training set, using different threshold t values and the fine-tuned parameters. This model labels triples taken from dataset U and returns a new augmented training set for each threshold;
3. we train and test once again the same learning method on the augmented training sets and compare the resulting MacroF$_1$ scores with the previous one in order to settle on the optimal t value.

We perform this evaluation with the training and test sets obtained from the labelled dataset L and with reduced training sets, i.e., having 1, 000, 500, and 200 triples.

4.1 LibLinear

For solving large-scale classification problems, we employ LibLinear (Fan et al., 2008), an open-source library for linear classification which supports logistic regression and linear support-vector machine, two popular binary linear classifiers.

Given a set of instance-label pairs (x_i, y_i), $i = 1, ..., l$, $x_i \in R^n$, $y_i \in \{-1, +1\}$, both methods solve the following unconstrained optimisation problem with different *loss functions* $\xi(w; x_i, y_i)$:

$$\min_w \ \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi(w; x_i, y_i)$$

where $C > 0$ is a penalty parameter selected separately for each learning method by performing 5-fold cross-validation on the original set L of manually labelled triples. Given training vectors $x_i \in R^n$, $i = 1, ..., l$ in two classes, and a vector $y \in R^l$ such that $y_i = \{1, -1\}$, a linear classifier generates a weight vector w as the model; the decision function is:

$$sgn(w^T x)$$

This means that a data point x is predicted as positive if $w^T x > 0$ and negative otherwise.

To obtain the best performances from logistic regression (LR) and linear support vector machines (SVM), it is necessary to tune their parameters, i.e., search for the optimal values for both the penalty term C and the threshold t for the confidence score of the prediction. Parameter C has been determined with 5-fold cross-validation among {1, 2, 4, 6, 8, 10, 16, 20, 25} values, while self-training has been tested with {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} as values of t; it is to be noted that the LibLinear SVC implementation does not return a confidence score of the prediction; hence we directly set $t = 0.0$.

As regards L2-regularised LR, from the 5-fold cross-validation, we obtained the following optimal C and $MacroF_1$ values, respectively: 2 and 0.8858 for the training set of 1, 600 triples, 6 and 0.8747 for the training set of 1, 000 triples, 20 and 0.8621 for the training set of 500 triples, 10 and 0.8610 for the training set of 200 triples. Table 2 illustrates the performance of LR after the self-training using fine-tuned parameters. Looking at the results, we can assume that the self-training strategy enhances OIE4PA performances when training with 1, 600, 1, 000, and 500 triples since the system achieves higher $MacroF_1$ scores; on the other hand, employing too few triples as in training set of 200 triples results in a worsening of the performance.

With respect to support-vector machines, from the 5-fold cross-validation, we obtained the following optimal C and $MacroF_1$ values respectively: 10 and 0.8793 for the training set with 1, 600 triples, 1 and 0.8753 for the training set with 1, 000 triples, 20 and 0.8674 for the training set with 500 triples, 2 and 0.8650 for the training set with 200 triples.

Conversely to the LR, when testing the self-training approach with a linear SVM classifier the results are overturned. Table 3 shows that the $MacroF_1$ decreases significantly when training on sets of 1, 600, 500, and 200 triples. We obtain an improvement in performance only with the training set of 1, 000 triples. Analysing results in Table 3, we observe an increase in P-0 and R-1 when the training size is 1,600, which probably indicates a slight overfitting since the model prefers the most frequent class (class 1).

4.2 XGBoost

We fine-tuned XGBoost parameters as well, that is to say, $Depth$ (maximum depth of a tree), Eta (learning rate), and $Round$ (number of rounds for boosting) on the training set made of 1,600 triples. We obtained the following optimal values and $MacroF_1$ score respectively: $Depth = 12$, $Eta = 0.4$, $Round = 80$ and 0.8234.

Table 2 $MacroF_1$ scores for optimal C (penalty term) and t (confidence score threshold) values with logistic regression after self-training

Set (C, t)	1, 600 (2, 0.7)	1, 000 (6, 0.7)	500 (20, 0.7)	200 (10, 0.2)
P-0	.9400	.9307	.9570	.8738
R-0	.7642	.7642	.7236	.7317
P-1	.9020	.9017	.8877	.8878
R-1	.9780	.9743	.9853	.9524
F_1-0	.8430	.8392	.8241	.7965
F_1-1	.9385	.9366	.9340	.9187
MacroF_1	.8908	.8879	.8790	.8576
Accuracy	.9116	.9090	.9040	.8838

Table 3 $MacroF_1$ scores for optimal C with support-vector machine after self-training

Set (C, t)	1, 600 (10, 0.0)	1, 000 (1, 0.0)	500 (20, 0.0)	200 (2, 0.0)
P-0	.9285	.8981	.9310	.8241
R-0	.7398	.7886	.6585	.7236
P-1	.8926	.9097	.8641	.8819
R-1	.9743	.9597	.9780	.9304
F_1-0	.8235	.8398	.7714	.7706
F_1-1	.9317	.9340	.9175	.9055
$MacroF_1$	.8776	.8869	.8444	.8380
Accuracy	.9015	.9065	.8787	.8661

Since SVC implementation does not return a confidence score of the prediction the value of t is set to 0.0

Table 4 depicts the performance of gradient boosting on the training set generated by self-training. The comparison of the values of $MacroF_1$ highlights how notably beneficial our self-training strategy proves to be, considering that it increments from 0.8234 to 0.8732.

4.3 BERT

We fine-tuned an Italian BERT model[19] using as input the triples and as output the two classes, 0: non-relevant and 1: relevant. We encode the input by using a max length of 512, while for the training we employed the Adam optimiser with a batch size of 16 and a learning rate of 0.00005 for 2 epochs. We train the model on the complete training dataset (1,600 triples) by using its 20% as validation set during the tuning. Afterwards, the model is evaluated on the test set. Results of the evaluation are reported in Table 5. We do not report results with different initial training sizes for the BERT classifier because, for this approach, we do not implement self-training.

The system is able to achieve similar results to LR but without using self-training.

4.4 Ablation test

In the field of Artificial Intelligence running an ablation test consists in removing a certain component of the system alternately, in our case a single feature, and verifying whether its absence improves or worsen the classifier performance. This study allows us to estimate the positive or negative impact of each feature on the classifying process.

We run ablation tests on the datasets which proved to obtain the best performance, thus the dataset composed of 1, 600 triples for the LR and the dataset composed of 1, 000 triples for the support vector machine, employing the same optimal parameters.

Tables 6 and 7 display the results of the ablation tests for LR and support vector machine, respectively. From a quick overview, we can state that the PoS-tags of subject, predicate, and object prove to be the feature with the highest impact on both LR and linear SVM since their $MacroF_1$ drops from 0.8908 to 0.8536 and from 0.8869 to 0.8462, respectively.

We conducted ablation tests on the training set of 1, 600 triples applying gradient boosting. By examining Table 8, we reach a conclusion in accordance with what we previously

[19] huggingface.co/dbmdz/bert-base-italian-uncased

Table 4 $MacroF_1$ score for optimal $Depth$, Eta, $Round$ and t values with gradient boosting after self-training

$Depth = 12$, $Eta = 0.4$, $Round = 80$, $t = 0.7$								
Set	P-0	R-0	P-1	R-1	F_1-0	F_1-1	MaF_1	Acc.
1, 600	.9109	.7419	.8930	.9674	.8177	.9287	.8732	.8975

observed: PoS-tags of the subject, predicate and object prove to be once more the most influential feature. On the contrary, we notice an unprecedented minor improvement in $MacroF_1$ when excluding the n-gram of the predicate feature.

Table 9 summarises the differences in the percentage of the ablation test for all the classifiers.

4.5 Search engine

In order to evaluate the search engine illustrated in Section 3.5, we set up an online assessment survey. We gathered 32 participants from Public Administration employees aged between 30 and 70 years, equally distributed by gender. The chosen sample considers final users of the proposed search engine with high familiarity with the domain of public tenders. We provided the participants in advance with a guide and then asked them to perform some searches and finally evaluate the search engine.

The survey presents a total of twelve questions: the first ten follow the questionnaire to measure the usability of our search engine, one open-ended question to provide detailed feedback on the system and their experience for suggesting improvements, and finally, the last question allows us to evaluate the likelihood the users would recommend a product, a service, or software.

According to the standard ISO 9241-11, usability can be measured in terms of system effectiveness, system efficiency, and system satisfaction. Created by John Brooke in 1986, the System Usability Score (Brooke et al., 1996) (SUS) proved to be intuitive and solid over hundreds of studies and nowadays, the SUS is widely used to measure the usability of websites and applications (Lewis, 2018). The survey consists of 10 questions with the 5-point Likert Scale (1 - Strongly Disagree, 5 - Strongly Agree):

Q1 *I think that I would like to use this website frequently.*
Q2 *I found the website unnecessarily complex.*
Q3 *I thought the website was easy to use.*
Q4 *I think that I would need the support of a specialist to be able to use this website.*
Q5 *I found the various functions in this website were well integrated.*
Q6 *I thought there was too much inconsistency in this website.*
Q7 *I would imagine that most people would learn to use this website very quickly.*
Q8 *I found the website very cumbersome to use.*
Q9 *I felt very confident using the website.*

Table 5 $MacroF_1$ score for the text classification model based on BERT

Set	P-0	R-0	P-1	R-1	F_1-0	F_1-1	MaF_1	Acc.
1, 600	.9073	.7854	.9081	.9660	.8420	.9362	.8891	.9075

Table 6 Results of the ablation tests on 1, 600 with the LR classifier

$C = 2, t = 0.7$	P-0	R-0	P-1	R-1	F_1-0	F_1-1	MaF_1	Acc.
Original	.9400	.7642	.9020	.9780	.8430	.9385	.8908	.9116
PoS-tags	.9140	.6911	.8746	.9707	.7870	.9201	.8536	.8838
pred. n-gram	.8624	.7642	.8989	.9450	.8103	.9214	.8659	.8888
dependencies	.9307	.7642	.9017	.9743	.8393	.9366	.8879	.9000
prev&post PoS	.9000	.7317	.8885	.9634	.8072	.9244	.8658	.8914
vector embed.	.8942	.7561	.8973	.9597	.8194	.9274	.8734	.8964

The first row shows the results of the classifier with all its features, while the other ones represent the results obtained by removing the indicated feature

Table 7 Results of the ablation test on a training set of 1, 000 triples with the SVM classifier

$C = 1, t = 0.0$	P-0	R-0	P-1	R-1	F_1-0	F_1-1	MaF_1	Acc.
Original	.8981	.7886	.9097	.9597	.8398	.9340	.8869	.9065
PoS-tags	.8775	.6992	.8758	.9560	.7783	.9142	.8462	.8762
pred. n-gram	.8448	.7967	.9107	.9341	.8201	.9222	.8712	.8914
dependencies	.8703	.7642	.8993	.9487	.8138	.9233	.8686	.8914
prev&post PoS	.8691	.7561	.8962	.9487	.8087	.9217	.8652	.8888
vector embed.	.8495	.7805	.9046	.9377	.8135	.9209	.8672	.8888

Table 8 Results of the ablation test on a training set of 1, 600 triples with XGBoost

$t = 0.0, Depth = 12, Eta = 0.4, Round = 80$	P-0	R-0	P-1	R-1	F_1-0	F_1-1	MaF_1	Acc.
Original	.8868	.7581	.8980	.9565	.8174	.9263	.8719	.8950
PoS-tags	.8660	.6774	.8680	.9529	.7602	.9085	.8343	.8675
pred. n-gram	.9029	.7500	.8956	.9638	.8194	.9284	.8739	.8975
dependencies	.8857	.7500	.8949	.9565	.8122	.9247	.8685	.8925
prev&post PoS	.8319	.7581	.8955	.9312	.7932	.9130	.8531	.8775
vector embed.	.8716	.7661	.9003	.9493	.8155	.9242	.8698	.8925

Table 9 Δ of the ablation tests for the three classifiers, i.e., logistic regression (LR), support vector machines (SVM) and XGBoost

	LR	SVM	XGBoost
Original	.8908	.8869	.8719
PoS-tags	- 4.26%	- 4.70%	- 4.40%
predicate n-gram	- 2.83%	- 1.80%	+ 0.24%
dependencies	- 0.32%	- 2.09%	- 0.39%
prev&post PoS	- 2.84%	- 2.48%	- 2.71%
vector embedding	- 1.97%	- 2.25%	- 0.23%

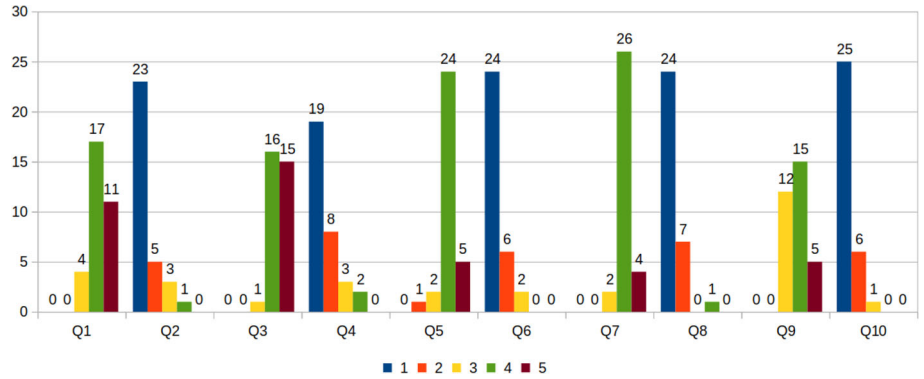

Fig. 5 Results for each question of the SUS questionnaire

Q10 *I needed to learn a lot of things before I could get going with this system.*

In order to compute the SUS score for each survey participant, we define $odd = (\sum odd_Q) - 5$ for the ratings given to every odd-numbered question, $even = 25 - \sum even_Q$ in the same manner, then $SUS = (odd + even) * 2.5$, which can assume values in the range [0, 100]. Open IE Search Engine SUS score, averaged on the number of participants, 32, is 83.98, considerably above the margin of the acceptable range, which guidelines (Sampaio, 2013) states to be 68. As illustrated in plot (a) of Fig. 6, the individual evaluation scores range between 60.0 and 97.5, with a median of 86.25. Figure 5 shows the responses to each of the ten questions. Considering the answers to Q1, Q3, Q7, and Q9 it can be affirmed that users found the proposed tool convenient and straightforward. This observation is backed up by the responses to even-numbered questions, negative-toned by definition, which predominantly gather around 1.

For an exhaustive evaluation, in addition to a numerical estimate, we asked the participants which features of the search engine we should improve through an open-ended question

Q11 *How could we improve our website?*

Table 10 contains the responses: the motif of their suggestions is to focus on the consistency of the presented information and documents, for example, by providing more details such as publication and due dates since they are vital data in the procurement domain.

Finally, in plot (b) of Fig. 6 displays the distribution of the answers to the Net Promoter Score question

Table 10 Answers to the open question "How could we improve our website?"

How could we improve our website?
I think it would be necessary to add examples/templates or an auto-complete function to the text bars. At first, it is not easy to understand what to search for.
Ensure an accurate and constant refresh of information given the limited life-cycles of open calls, competitions, and tenders.
Add the publication date of the notice to see how up-to-date the information is.
Improve the graphic appearance.
I cannot assess whether the information is up to date or not.

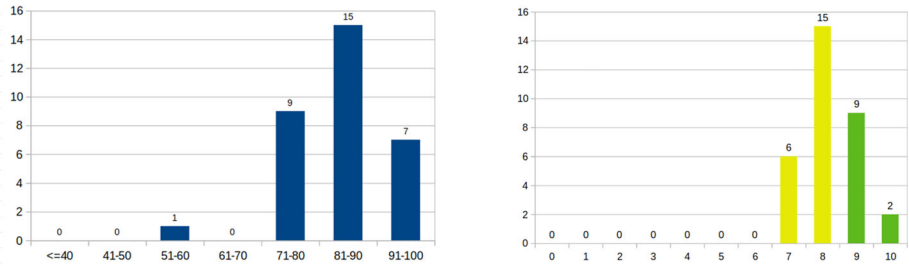

Fig. 6 (a) Distribution of the SUS scores. (b) Answers to "How likely is it that you would recommend Open IE Search Engine to a friend or colleague?"

Q12 *How likely is it that you would recommend Open IE Search Engine to a friend or colleague?*

On a scale from 0 to 10, the 18.8% of the participants answered 7, 46.9% 8, 28.1% 9 and 6.3% 10. The idea behind the NPS is to divide the users into *promoters*, *passives* and *detractors* of the item, based on their answer: users providing ratings between 10 and 9 are considered to be promoters, between 8 and 7 are passives and finally, from 6 to 0 are detractors. The NPS is computed as follows the number of *promoters* minus the number of *detractors* and can assume values included in the interval $[-100, +100]$; despite the fact that the computation occurs between percentages, the NPS is actually expressed as a decimal value. General guidelines established by Bain&Co.,[20] inventors of the NPS, state that any positive, non-zero score of the NPS is considered "good", since it means that there are more promoters than detractors; however, any score above 20 is considered encouraging, whereas 50 is excellent and above 80 first-rate. Our Open IE Search Engine scores 34.4, having 34,4% of promoters (28.1% voted 9, 6.3% voted 10) and 0% detractors.

4.6 Dataset and software

All the code necessary to run experiments and build the index for the search engine is available on GitHub[21]. The code of the search engine web interface is also available on GitHub[22]. The dataset is composed of:

- the corpus of 6,262 texts extracted from Italian public tenders;
- the training set of 1, 600 annotated triples;
- the test set of 400 annotated triples;
- the set U of 14, 096 triples used for the self-training;
- a compressed archive that contains both the extracted triples and the index for each supervised approach.

The dataset is available here[23].

[20] www.bain.com/insights/introducing-the-net-promoter-system-loyalty-insights/

[21] github.com/pippokill/WikiOIE

[22] github.com/pippokill/oie-web

[23] https://doi.org/10.5281/zenodo.8331106

5 Implications of research, limits, and challenges

OIE4PA, the Open Information Extraction tool for tenders described in this paper is able to achieve good results, as proven by the outcomes of the in-vitro evaluation. Additionally, the search engine based on the extracted triples proves to be effective, according to the in-vivo evaluation. The promising results of the in-vivo evaluation allowed us to assess not only the performance of the search engine and the quality of the triples extracted by OIE4PA, but also show how Open IE algorithms can find their employment in the public tender sector. Documents created through the whole e-procurement cycle are often very verbose and can consist of thousands of pages. For this reason, Open IE systems can be used to extract relevant information that can help Project Managers perform their daily activities.

The proposed methodology combines traditional classification algorithms with a self-training approach to handle the data scarcity problem. Moreover, we propose for the first time a dataset of manually annotated relevant and non-relevant triples from a corpus of Italian tenders. With respect to the limitations, more work can be done to increase the size of the dataset. A large dataset is a necessary condition for the adoption of methodologies based on deep learning techniques, especially the generative ones based on the text-to-text generation that has proven to be effective in the information extraction task. We believe that an adaptation of the search engine can help to collect more relevant triples through the users' feedback to increase training data. One of the main challenges is not knowing the a priori set of relations we want to extract in our application domain. Generally, the relevant relations differ from those found in common knowledge resources such as Wikipedia, DBpedia, or Wikidata. In addition, relevant relations may change from tenders to tenders. This means that it is impossible to automatically build training data from these resources, useful for our domain. Therefore the users' feedback during the use of the search engine can be exploited to understand the relations the users are more interested in.

6 Conclusions and future works

In this paper, we have presented OIE4PA, an Open IE framework that can identify facts from Public Administration documents, such as tenders. OIE4PA has been trained over the Italian language, however, thanks to its architecture, it can be adapted to cover different languages, too. By exploiting the data extracted from an official PA website, we developed a dataset consisting of 6,262 documents. Domain experts labelled as *relevant* or *non-relevant* a partition containing 2,000 of these tuples, from which we created the training and test sets. These manually labelled datasets allowed us to train our model on this specific domain and evaluate its performance in terms of Precision, Recall and Macro average F_1. Three learning models, i.e., logistic regression, linear support-vector machine and gradient boosting, were trained on the aforementioned training set and on an additional *augmented* training set obtained with a self-training approach. We compared the performances of OIE4PA with both training sets to verify whether the self-training approach enhances them or not. As described in Section 4, OIE4PA obtains promising results after the self-training step, achieving a MacroF_1 0.8858 to 0.8908 with logistic regression, from 0.8793 to 0.8869 with support-vector machine, from 0.8234 to 0.8732 with gradient boosting. OIE4PA was also positively evaluated through an in-vivo experiment that directly involved experts in the PA sector.

In future work, we plan to take into account the feedback given by the users for enhancing the existing interface and the quality of triples prediction. Our idea is to implement a human-

in-the-loop strategy in which expert users can help to improve the model quality, for example, by annotating wrongly classified triples directly through the user interface. Furthermore, we intend to exploit the triples extracted from the documents for a wider variety of applications, e.g., guiding the creation of short synopsis of each document through text summarisation techniques or visualising the extracted triples via graphs, which would allow navigating among documents in a more interactive way. This will also help develop a more advanced version of our prototype, which will be proposed to more users for a broader study.

Acknowledgements We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 6 - Symbiotic AI (CUP H97G22000210007) under the NRRP MUR program funded by the NextGenerationEU

Author Contributions **Lucia Siciliani**: Conceptualisation, Methodology, Software, Validation, Investigation, Writing - original draft, Writing - review & editing.
Pierpaolo Basile: Conceptualisation, Methodology, Software, Validation, Investigation, Writing - original draft, Writing - review & editing, Supervision, Project administration.
Eleonora Ghizzota: Conceptualisation, Methodology, Software, Validation, Investigation, Writing - original draft, Writing - review & editing.
Pasquale Lops: Conceptualisation, Writing - original draft, Writing - review & editing, Supervision.

Funding Open access funding provided by Università degli Studi di Bari Aldo Moro within the CRUI-CARE Agreement. Open access funding is provided by Università degli Studi di Bari Aldo Moro within the CRUI - 'CARE Agreement'.

Availability of data and materials The dataset is publicly available on Zenodo.

Code Availability The code that supports the findings of this study as well as the code of the search engine web interface, is publicly available on open platforms.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

Competing interests The authors have no financial or proprietary interests in any material discussed in this article.

Ethics approval We declare that this submission follows the policies as outlined in the Guide for Authors.

Consent to participate The evaluation process does not store sensitive information about participants, and survey answers cannot be attributed to any specific individual.

Consent for publication All authors agree with the content and give explicit consent to submit.

References

Amini, M., Feofanov, V., Pauletto, L., et al. (2022). Self-training: A survey. arXiv:2202.12040

Banko, M., Cafarella, M.J., Soderland, S., et al. (2007). Open Information Extraction from the Web. In: Veloso MM (ed) IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007, pp. 2670–2676, http://ijcai.org/Proceedings/07/Papers/429.pdf

Barthélemy, F., Ghesquière, N., Loozen, N., et al. (2022). *Natural language processing for public services*. Luxembourg: Publications Office of the European Union.

Bojanowski, P., Grave, E., Joulin, A., et al. (2016). Enriching Word Vectors with Subword Information. arXiv preprint arXiv:1607.04606

Brooke, J., et al. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry, 189*(194), 4–7.

Buchholz, S., Marsi, E. (2006). CoNLL-X Shared Task on Multilingual Dependency Parsing. In: Màrquez, L., Klein, D. (eds). Proceedings of the Tenth Conference on Computational Natural Language Learning, CoNLL 2006, New York City, USA, June 8-9, 2006, pp. 149–164. https://aclanthology.org/W06-2920/

Cabot, P. H., & Navigli, R., et al. (2021). REBEL: relation extraction by end-to-end language generation. In M. Moens, X. Huang, & L. Specia (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event/Punta Cana, Dominican Republic (pp. 2370–2381). Dominican Republic. https://doi.org/10.18653/v1/2021.findings-emnlp.204

Cassotti, P., Siciliani, L., Basile, P., et al. (2021). Extracting Relations from Italian Wikipedia using Unsupervised Information Extraction. In: V.W. Anelli, T.D. Noia, N. Ferro, et al. (eds). Proceedings of the 11th Italian Information Retrieval Workshop 2021, Bari, Italy, September 13-15, 2021. http://ceur-ws.org/Vol-2947/paper2.pdf

Chen, T., Guestrin, C. (2016). Xgboost: A scalable tree boosting system. arXiv:1603.02754

Christensen, J., Soderland, S., Etzioni, O., et al. (2010). Semantic role labeling for open information extraction. In: Proceedings of the NAACL HLT 2010 first international workshop on formalisms and methodology for learning by reading, pp 52–60

Christensen, J., Mausam, Soderland, S., et al. (2011). An analysis of open information extraction based on semantic role labeling. In: M.A. Musen,Ó. Corcho (eds) Proceedings of the 6th International Conference on Knowledge Capture (K-CAP 2011), June 26-29, 2011, Banff, Alberta, Canada, pp 113–120, https://doi.org/10.1145/1999676.1999697

Corro, L. .D., & Gemulla, R., et al. (2013). ClausIE: clause-based open information extraction. In D. Schwabe, V. .A. .F. Almeida, & H. Glaser (Eds.), *22nd International World Wide Web Conference, WWW '13* (pp. 355–366). https://doi.org/10.1145/2488388.2488420

Damiano, E., Minutolo, A., & Esposito, M., et al. (2018). Open Information Extraction for Italian Sentences. In L. Barolli, M. Takizawa, & T. Enokido (Eds.), *32nd International Conference on Advanced Information Networking and Applications Workshops* (pp. 668–673). AINA 2018 workshops. https://doi.org/10.1109/WAINA.2018.00165

Dognin, P., Padhi, I., Melnyk, I., et al. (2021). ReGen: Reinforcement learning for text and knowledge base generation using pretrained language models. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic, (pp. 1084–1099), https://doi.org/10.18653/v1/2021.emnlp-main.83

Dwivedi, Y. K., Hughes, L., Ismagilova, E., et al. (2021). Artificial intelligence (ai): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. *International Journal of Information Management, 57*, 101994.

Etzioni, O., Fader, A., Christensen, J., et al. (2011). Open Information Extraction: The Second Generation. In T. Walsh (Ed.), *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence* (pp. 3–10). https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-012

Fan, R., Chang, K., Hsieh, C., et al. (2008). LIBLINEAR: A library for large linear classification. J Mach Learn Res 9:1871–1874. https://dl.acm.org/citation.cfm?id=1442794

Gashteovski, K., Gemulla, R., Kotnis, B., et al. (2020). On aligning OpenIE extractions with knowledge bases: A case study. In: *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, Online, (pp. 143–154), https://doi.org/10.18653/v1/2020.eval4nlp-1.14

Guarasci, R., Damiano, E., Minutolo, A., et al. (2020). Lexicon-Grammar based open information extraction from natural language sentences in Italian. *Expert Syst Appl, 143*,. https://doi.org/10.1016/j.eswa.2019.112954

Josifoski, M., De Cao, N., Peyrard, M., et al. (2021). GenIE: generative information extraction. arXiv:2112.08340

Kalampokis, E., Karacapilidis, N., Tsakalidis, D., et al. (2023). *Understanding the use of emerging technologies in the public sector: A review of horizon 2020 projects*. Digital Government: Research and Practice. https://doi.org/10.1016/j.eswa.2019.112954

Kenton, J.D.M.W.C., Toutanova, L.K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of NAACL-HLT*, (pp. 4171–4186)

Landis, J.R., Koch, G.G. (1977). The measurement of observer agreement for categorical data. Biometrics (pp. 159–174)

Lewis, J. R. (2018). The system usability scale: past, present, and future. *International Journal of Human-Computer Interaction, 34*(7), 577–590.

Madan, R., & Ashok, M. (2023). AI adoption and diffusion in public administration: A systematic literature review and future research agenda. *Gov Inf Q, 40*(1), 101774. https://doi.org/10.1016/j.giq.2022.101774

Mausam, Schmitz, M., Soderland, S., et al. (2012). Open Language Learning for Information Extraction. In: J. Tsujii, J. Henderson, M. Pasca (eds) *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea, (pp. 523–534). https://aclanthology.org/D12-1048/

Niklaus, C., Cetto, M., Freitas, A., et al. (2018). A Survey on Open Information Extraction. arXiv:1806.05599

Sampaio, A. (2013). Quantifying the user experience: Practical statistics for user research by jeff sauro and james r. lewis. *SIGSOFT Softw Eng Notes, 38*(1), 57–58. https://doi.org/10.1145/2413038.2413056

Siciliani, L., Cassotti, P., Basile, P., et al. (2021). Extracting Relations from Italian Wikipedia using Self-Training. In: E. Fersini, M. Passarotti, V. Patti (eds) *Proceedings of the Eighth Italian Conference on Computational Linguistics*, CLiC-it 2021, http://ceur-ws.org/Vol-3033/paper28.pdf

Straka, M., & Straková, J. (2017). Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with udpipe. In J. Hajic & D. Zeman (Eds.), *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* (pp. 88–89). https://doi.org/10.18653/v1/K17-3009

Vo, D., Bagheri, E. (2016). Open Information Extraction. arXiv:1607.02784

Wu, F., Weld, D.S. (2010). Open Information Extraction Using Wikipedia. In: J. Hajic, S. Carberry, S. Clark (eds) *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, July 11-16, 2010, Uppsala, Sweden, (pp. 118–127), https://aclanthology.org/P10-1013/

Zhu, J., Nie, Z., Liu, X., et al. (2009). StatSnowball: a statistical approach to extracting entity relationships. In J. Quemada, G. León, Y. .S. Maarek, et al. (Eds.), *Proceedings of the 18th International Conference on World Wide Web, WWW 2009.* . https://doi.org/10.1145/1526709.1526724