Toward a Human-in-the-Loop Approach to Create Training Datasets for RDF Lexicalisation

Jessica Amianto Barbato[1(✉)], Marco Cremaschi[1], Anisa Rula[2], and Andrea Maurino[1]

[1] University of Milano-Bicocca, Milan, Italy
{JessicaAmianto.Barbato,Marco.Cremaschi,Andrea.Maurino}@unimib.it
[2] University of Brescia, Brescia, Italy
anisa.rula@unibs.it

Abstract. Datasets that include alignments between natural language and Knowledge Graphs are fundamental to a wide variety of Natural Language Processing and Generation tasks. Current state-of-the-art aligned datasets, though, are significantly impacted by reduced size and scarcity of covered domains, and their quality is difficult to evaluate. To compensate for these issues, we introduce SEALIon, a tool for extracting RDF triples from natural language textual corpora based on a human-in-the-loop approach. We present our first results of SEALIon's approach, paving the way for further researches in the field of human-in-the-loop triple extraction.

Keywords: Natural language processing · Natural language generation · Human-in-the-loop · Relation extraction

1 Introduction

Handling human knowledge during the creation of high-quality resources is of crucial importance for many research tasks related to Natural Language Processing (NLP), such as Question Answering (QA), Knowledge Base Population (KBP) [10], Entity Linking (EL), Named Entity Recognition (NER) and Natural Language Generation (NLG). As a prerequisite of these tasks, the alignments between natural language and structured Knowledge Graph (KG) are fundamental for the training of Machine Learning (ML) approaches. Such alignments can be briefly described as an inter- or cross-document association between a sentence as shown in Example 1 and its representation in Resource Description Framework (RDF) triples as shown in Example 2. However, the datasets of this kind that are currently available suffer from several issues related primarily to their reduced size, the narrowness of the domain they address, and/or the quality of the results obtained [7].

Example 1. "Elliott" Smith was an American singer, songwriter, and multi-instrumentalist.

K. Arai (Ed.): IntelliSys 2023, LNNS 822, pp. 84–101, 2024.
https://doi.org/10.1007/978-3-031-47721-8_6

Example 2. ⟨ dbr:Elliott_Smith, dbo:occupation, dbr:Songwriter ⟩, ⟨ dbr:Elliott_Smith, dbo:occupation, dbr:Singer ⟩, ⟨ dbr:Elliott_Smith, dbo:occupation, dbr:Multi-instrumentalist ⟩.

Several works have been provided for the creation of aligned datasets, which consider two main operations: NER and Relation Extraction (RE). RE via distant and weak supervision, regardless of how inclusive is the fundamental assumption they are based on, are proven to produce rather noisy data [20,23,26,27]. Even though the distant supervision assumption is largely relaxed in our approach, it would be naive to consider that our resulting data will not be noisy as well: this explains why a fruitful denoising approach is necessary in order to improve the quality of such data. Relation extraction is a challenging task for computers to complete fully automatically, mostly due to tasks like entity extraction and predicate linking. Therefore triple alignment could easily benefit from intellectual crowds [3]. Crowdsourcing has been efficiently employed to address information extraction and data labelling tasks by employing crowds' abilities in disparate domains. Moreover, the public availability of crowdworking platforms like Amazon Mechanical Turk (AMT)[1] [1,20] and Figure Eight[2] (formerly known as CrowdFlower) [9,21] has eased the process of collecting data from crowdworkers.

Following the intuitions provided in [14], there are several important issues related to crowdsourced data processing: (i) **Task design**: previous works that leverage crowdsourcing in a human-in-the-loop approach to information extraction present a worker with a single choice scenario, in which they have either to decide whether a sentence expresses a target relation [15] or pick one possible label for each portion of text in a sentence [12]; (ii) **Quality control**: spiteful workers could deliberately give the wrong answers, and improperly trained workers could not complete the annotation task successfully; (iii) **Cost control**: many examples to be annotated, even at very low retribution per worker and task, correspond to a relevant monetary effort on the part of the requester. Human-in-the-loop approaches are inherently less impacted by cost-related issues since the amount of tasks submitted to each crowdworker is a small remainder of the tasks that cannot be automatically completed by the system; (iv) **Latency control**: when the task is too difficult, not appealing, nor interesting, crowdworkers might take too much time to complete their job. It could be tempting, though, to raise the retribution in order to reduce the latency, but it might not always be the best solution [8].

In this work, we propose a human-in-the-loop approach to produce high-quality aligned RDF-text datasets. When referring to *human-in-the-loop* with respect to our approach, a process will be outlined in which manually revised input is included within an RDF annotation and triple extraction process in which both tasks are performed automatically. We present SEALIon (SEmantic

[1] https://www.mturk.com/.
[2] https://appen.com/.

ALIgnment),[3] an annotation tool to assist the users. Our approach is different with respect to previous approaches that exploit human-in-the-loop and from those systems that use active learning. Indeed, in these cases, the user is mostly asked to exhibit agreement or a preference concerning an output provided by the system. Moreover, SEALIon is able to limit the issues due to the quality of alignments and the narrowness of the reference domain, highlighted in [7].

The main contributions of this paper are as follows: (i) A systematic analysis of the state-of-the-art of the RDF-text datasets to highlight their characteristics and creation process; (ii) The definition of a pipeline for the generation of RDF-text datasets characterised by high-quality alignments; (iii) The introduction of a crowdsourcing approach for the relation extraction process; (iv) A user-friendly Web app that allows users to validate the alignments; (v) A first empirical evaluation of the proposed workflow.

The remainder of the paper is organised as follows: Sect. 2 presents an overview of currently available aligned datasets and their significant characteristics; Sect. 3 introduces SEALIon's approach by presenting the RDF extraction pipeline it is based on; Sect. 4 provides a detailed description of how the crowdsourcing module is designed; Sect. 5 discusses the assessment of SEALIon's results in terms of both automatic and manual evaluation. Eventually, Sect. 6 formalises the main contributions of this work.

2 State of the Art

State-of-the-art open-domain datasets employed in data-driven NLG and other applications are T-REx [7], CC-DPB [10] and the WebNLG Dataset [4,9,24].

T-REx is one of the broadest and most recent among the publicly available open-domain training datasets for NLG tools; it features 6.2 million sentences from DBpedia abstracts aligned with 11 million triples from Wikidata. It has been previously employed, fully or partially, for testing Named Entity Linking (NEL) approaches, language models, unsupervised models for relation and information extraction, self-supervised frameworks for Open Relation Extraction and Knowledge Base Population systems; it has also been employed for training and testing analogy models for learning relations [13,16,25].

CC-DBP aligns 173 million sentences from the Common Crawl[4] corpora with data from DBpedia, resulting in 3 million aligned triples. Common Crawl is a valuable source of open-domain textual contents, but the model that produced CC-DBP can only account for small portions of such texts. The dataset has been mostly employed to evaluate KBP systems [11] and for the training and testing steps of transfer learning systems development [2].

The **WebNLG** dataset has been employed to complete the 2017 WebNLG challenge (updated in late 2019 and re-released in 2020 for the 2020 WebNLG+ challenge), and it has been used in the development and experimental steps for

[3] The version 0.1 of the tool is available at https://sealion.ml/. The source code can be downloaded from the Git repository https://bitbucket.org/disco_unimib/sealion/.

[4] https://commoncrawl.org/.

NLG from structured data. As per the original version, the English dataset contains 13,211 entries with 35,426 lexicalisations and 372 unique RDF properties in the training set and 1,667 entries with 4,464 lexicalisations and 290 distinct RDF properties in the development set; the RDF triples in the set belong to fifteen DBpedia categories, including WrittenWorks, Artist, Athlete and Food. Note that the WebNLG dataset includes original and modified DBpedia triples, the former featuring only DBpedia predicates and the latter containing more human-understandable relations; the more recent 3.0[5] version includes one more DBpedia category for a total of 16 topics and the syntactic shape of each sentence. It has been created by leveraging a novel method that combines a content selection module to extract varied, relevant and coherent data units from DBpedia with a crowdsourcing process for associating data units with human authored textual content to capture their meaning. The dataset is far more precise when compared to others in both entity and relation extraction, and it features both multiple lexicalisations of the same triple and multiple triples that can be aligned to one, more complex, sentence; still, it is rather restricted as for both the length and intricacy of the sentences and the domain of interest. Interesting examples of neural network models trained on the WebNLG dataset are present in literature [4], but they only apply to the restricted range of domains featured in the dataset; moreover, the structural complexity of such output texts is not comparable to human-produced natural language corpora.

In Table 1, it is possible to view three examples extracted from the datasets described above. It can be seen that there are significant differences between WebNLG, T-REx, and CC-DBP. In WebNLG, the sentences have a much simpler structure, and there is a close correspondence between the information contained in the sentence and the information expressed by the triples. In T-REx and CC-DBP, on the other hand, the triples do not express all the information contained in the sentences, which have a more complex structure.

As previously indicated, the three datasets were created to enable different tasks: T-REx for RE and KBP, CC-DBP to evaluate KBP systems and for the training and testing transfer learning systems. WebNLG for training NLG systems.

Related to NLG, many approaches have been proposed for the last twenty years and can be divided into two main categories: **pipeline** and **end-to-end** systems [5]. The former have better performances but are more complex: they generally have a complex architecture composed of several modules; this kind of architecture is prone to error propagation, however, it is less susceptible to hallucinations, which happen when the system adds information that is not present in the input (basically, when they say things that are not true) [5]. The latter, the **end-to-end** systems, have much simpler architectures but are less precise: they are more prone to hallucinations, omissions and repetitions. Nevertheless, they excel in some tasks such as discourse ordering and text structuring [5]. Recent end-to-end approaches such as T5[6] [22] have closed the gap between the

[5] Available at https://gitlab.com/shimorina/webnlg-datas.
[6] https://github.com/google-research/text-to-text-transfer-transformer.

Table 1. Examples of alignments contained in the analysed datasets

Dataset	Triple	Sentences
T-REx	⟨Ljubljana, country, Slovenia⟩ ⟨Ljubljana, capital of, Slovenia⟩ ⟨Slovenia, capital, Ljubljana⟩ ⟨Ljubljana, capital of, Slovenia⟩	Jason has connections outside the classical world, being the mythical founder of the city of Ljubljana,the capital of Slovenia.
CC-DBP	dbr:The_Beatles, dbr:Tony_Sheridan ⟨dbo:associatedBand, ⟨odp:isMemberOf, ⟨dbo:associatedMusicalArtist, ⟨odp:hasMember	Ain't She Sweet was an American album featuring four tracks recorded in Hamburg in 1961 by The Beatles featuring Tony Sheridan (except for the title ...
WebNLG	⟨Aarhus, leaderName, Jacob_Bundsgaard⟩ ⟨Aarhus_Airport, city, Aarhus⟩	Aarhus airport serves the city of Aarhus whose leader is Jacob Bundsgaard.

two architectures, surpassing pipelines in many tasks and showing excellent generalisation capabilities but still suffering from problems such as hallucinations, omissions and repetitions. On the other hand, pipeline systems like DualEnc [28] have demonstrated the importance of relations between the triples to create semantically correct sentences, which can be represented using graph structures such as Knowledge Graphs (KGs). This analysis suggests that, even if progress has been made in language generation approaches, these are closely related to the dataset's quality used in the training phase.

Other approaches of RE in the literature yield datasets whose quality is comparable to that of CC-DBP [10]. This entails that such datasets are unlikely to be employed in an NLG context like the one outlined here. In contrast, the goal of this work aims to achieve a quality that can be approached to that of WebNLG without relying on a major use of manual annotation.

3 Defining a Pipeline for RDF Extraction from Text

We introduce a pipeline for RDF triple extraction from text, as shown in Fig. 1, which follows in the footsteps of the process described in [7].

Note that the pipeline in [7] has been fully re-implemented to make several adjustments aimed at improving the overall functionality of the system. Key

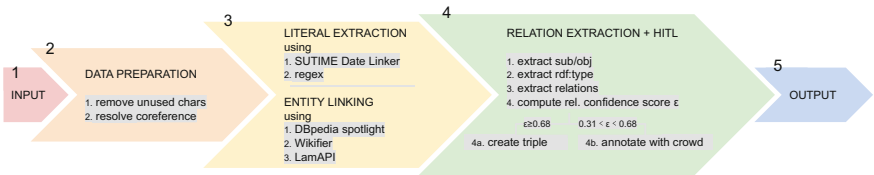

Fig. 1. The SEALIon pipeline described in Sect. 3

components of the pipeline are presented in the following sections. One of the most significant differences is modifying the process to consider a user intervention and adding User Interface elements to provide appropriate support to the User Experience during the annotation tasks.

3.1 Data Import and Preparation

The first step in our pipeline is **data import**; as a matter of convenience, the pipeline is fed with text (see Example 3).

Example 3. "Elliott" Smith was a singer. Elliott Smith was born in 1969. He lived in Portland, Oregon.

When a textual corpus has been uploaded, the application performs some **pre-processing** tasks including: (i) **Removal of unused characters** refers to the application of regular expressions to remove unusable portions of the text such as asides between hyphens and dashes, and phonetic transcriptions to improve the efficacy of the system; (ii) **Coreference resolution** leverages spaCy[7] and Stanford CoreNLP[8] through API calls to resolve implicit subjects in the text. An implicit subject is found in a sentence when the agent of the action is not directly expressed. Along with POS tagging, it is necessary to differ subject from object entities in the text.

The output of the pre-processing step is exemplified in Example 4; if compared to the same, raw input in Example 3, we notice that some portions of text (e.g., birthplace and date between parenthesis in the first sentence) have been removed and implicit subjects ("he") have been replaced by the actual subject *Elliott Smith*, underlined in the example.

Example 4. Elliott Smith was a singer. Elliott Smith was born in 1969. Elliott Smith lived in Portland, Oregon.

3.2 Entity Resolution

The second step in SEALIon's pipeline is **entity resolution** which consists of two tasks: (i) **Entity Linking**: Wikifier's and DBpedia Spotlight's[9] APIs are employed; both services are capable of providing automatic annotations of entities in the text with links to either Wikipedia concepts or DBpedia entities. Further support to the entity linking process is provided by the LamAPI[10] tool, which supports fast queries for DBpedia and Wikidata using an ElasticSearch instance. A set of annotated entities for each service is produced, the union of which will constitute the pool of entities present in the text. Then, the same services are used to retrieve the classes to which the entities belong.; (ii) **Literal**

[7] https://spacy.io/.
[8] https://stanfordnlp.github.io/CoreNLP/.
[9] https://www.dbpedia-spotlight.org/api.
[10] https://lamapi.ml/.

extraction: literals in text are extracted with the help of SUTime Date Linker[11] and regular expressions to identify geo-coordinates, URLs, e-mail addresses, IP addresses and ISBNs [6]. The result of the entity resolution is shown in Example 5.

Example 5. **dbr:Elliott_Smith** was a singer. **dbr:Elliott_Smith** was born in **1969**. **dbr:Elliott_Smith** lived in **dbr:Portland,_Oregon**.

After entity resolution is completed, subject and object entities, along with literals, are identified in the text; given a document $D = (S_0, S_1, S_2, ..., S_z)$ with $0 \leq n \leq z$ where S_n is a sentence in the document D, in input similar to the one in Example 3, we obtain a set of tuples $t_{n,m} = (sub_{n,m}, obj_{n,m})$ with $t_{n,m}$ in the sentence S_n, each representing the m-th couple of subject and object entities in the n-th sentence. For clarity, we provide an example of the output of the current module (see Example 6), which will be sent to the relation extraction step.

Example 6. **Sentence**: Elliott Smith was a singer. Elliott Smith was born in 1969. Elliott Smith lived in Portland, Oregon
Tuples: (dbr:Elliott_Smith, dbr:singer),(dbr:Elliott_Smith,xsd:date),
(dbr:Elliott_Smith, dbr:dbr:Portland,_Oregon)

3.3 Relation Extraction

Given a document D and a set of tuples of type $t_{n,m} = (sub_{n,m}, obj_{n,m})$, the **relation extraction** process is aimed at identifying predicates that relate those entities in the context of the sentence they belong to. Note that we assume that each sentence might be represented by more than one triple, but we also admit the case in which no triple is extracted that can express the content of the sentence. Our relation extraction process is based on three different steps: (i) **Step 1**: looks for candidate predicates employing the LamAPI tool. The result of this step is a list of the top 5 most suitable predicates; (ii) **Step 2**: identifies the correct predicate by means of a set of syntactic and semantic metrics and appropriate thresholds. (iii) **Step 3**: activates the human-in-the-loop step for ambiguous situations.

The input in ***Step 1*** consists of a pool of tuples of type $t_{n,m} = (sub_{n,m}, obj_{n,m})$ for each sentence $S_n \in D$. For each tuple, predicate linking is performed by leveraging LamAPI, with which it is possible through an API specification to match labels with full-text search capability and fuzzy matching. With the terms `SubjectType` and `ObjectType` we refer to the ontology class (e.g., rdf:type) or a datatype, so that the aforementioned triple comes to represent the presence of instances of two given classes that are linked to each other via properties `predicate`. With LamAPI, we perform a first query to retrieve the predicates $pred_{n,m}$ between $sub_{n,m}$ and $obj_{n,m}$. A second query permits to obtain an ordered set (based on frequency) of at most five predicates $pred_{n,m,j}$

[11] https://stanfordnlp.github.io/CoreNLP/sutime.html.

between $typeSub_{n,m}$ and $typeObj_{n,m}$ where n is the sentence in the document D, m is the tuple $t \in S_n$ and $0 \leq j \leq 4$ is the predicate retrieved from LamAPI.

In ***Step 2*** each predicate is compared to the predicate $pred_{S_{n,m}}$ in the sentence. To assess the similarity of each relation $pred_{n,m,j}$ to the predicate $pred_{S_{n,m}}$ the following similarity metrics are applied: (i) **String matching**: we look for a string matching between the RDF predicate and a portion of text. Note that we are not limiting the search to predicates since many RDF relations are made up of one or more nouns and adjectives; (ii) **Synonyms string matching**: a set of synonyms for the RDF predicate is retrieved via RelatedWords,[12] a service for retrieving word or n-grams synonyms; (iii) **Word embeddings similarity**: we use Word2Vec [18,19] to generate word embeddings, which are vectorial representations, and compute similarities. Unlike the string matching comparison, we need to limit our search to predicates in the sentence because the process would be otherwise too expensive in terms of computation, and we might identify similar words that are not representative of any relation between a couple of entities. To do so, we employ part-of-speech tags to extract a predicate from the text and compute its cosine similarity to the RDF predicate.

As for the word embedding similarities, a point has to be made; imagine that the verb in a sentence $S_n \in D$ is phrasal or that the predicate we have collected from LamAPI is multi-word. In either situation, we would not be able to compute a single vector that expresses the meaning of that relation since Word2Vec can only calculate single-word vectors. We are now presented with two different scenarios: (i) The predicate $pred_{n,m,j}$ in the triple has multiple words (e.g., dbo:associatedMusicalArtist): in this case, we create a single vector for each word in $pred_{n,m,j}$ and then we compute the average of those embeddings. We are here assuming that the meaning of the words in the predicate are similar, otherwise, the average vector would be meaningless; (ii) The predicate $pred_{S_{n,m}}$ in the sentence is phrasal or multi-word (e.g., "bring out"): phrasal verbs cannot be vectorialised by computing an average of its components since lexical vector representations of particles like "out" or "up" do not significantly participate in distinguishing different phrasal verbs derived from the same verb (e.g., "bring out" and "bring up"). We decided to replace each phrasal verb with a single-word verb collected by leveraging WordNet[13] synsets and then compute its embedding; by doing so, we can calculate the cosine similarity between $pred_{S_{n,m}}$ and $pred_{n,m,j}$.

A single value associated to each predicate $pred_{n,m,j}$ is returned so that the likelihood $L_{pred_{n,m,j}}$ of it representing the relation expressed in the sentence can be assessed; the highest scoring predicate $\max_{L_{pred_{n,m,j}}}$ is selected for triple creation. We compute two thresholds ε_{min}, $\varepsilon_{max} \in (0, 1)$ with $\varepsilon_{min} < \varepsilon_{max}$ so that:

- if $\max_{L_{pred_{n,m,j}}} \leq \varepsilon_{min}$ the triple $T_{n,m,j}$ is discarded and no triple is returned for sentence S_n;

[12] https://relatedwords.org/.
[13] https://wordnet.princeton.edu/.

– if $\varepsilon_{min} < \max_{L_{pred_{n,m,j}}} < \varepsilon_{max}$ the triple $T_{n,m,j}$ is sent to selected crowdworkers for manual annotation;
– if $\max_{L_{pred_{n,m,j}}} \geq \varepsilon_{max}$ the triple $T_{n,m,j}$ is created and stored.

Because the selection of thresholds is highly dependent on the corpus chosen for Word2Vec training, we specify that in the validation process, the Brown corpus, available among the Python NLTK library corpora, was chosen to initialize the model. The thresholds have been thus empirically set to $\varepsilon_{min} = 0.425, \varepsilon_{max} = 0.645$. Consider the sentence and the tuples: dbr:Elliott_Smith, dbr:Bass_Guitar and dbr:Elliott_Smith, dbr:Guitar in Example 6. The sentence can be split into two verb phrases:

Example 7. Sentence: Elliott Smith's primary instrument was the guitar
Tuple: (dbr:Elliott_Smith, dbr:Guitar) converted to (dbo:MusicalArtist, dbo:MusicalArtist)

Example 8. Sentence: Elliott Smith also played piano, clarinet, bass guitar, drums, and harmonica
Tuple: (dbr:Elliott_Smith, dbr:Bass_Guitar) converted to (dbo:MusicalArtist, dbo:MusicalArtist)

Examples 7 and 8 show the alignments between the two sentences and the tuples that represent their subject and object. We can now query LamAPI with the converted tuples; note that the tuples containing each entity's types are equal, so the output of the query will be the same, as described in Example 9.

Example 9. Predicates: dbo:associatedMusicalArtist, dbo:associatedBand, dbo:recordLabel, dbo:instrument, dbo:influencedBy

A value $L_{pred_{n,m,j}}$ for each $pred_{n,m,j}$ returned by LamAPI is computed following the aforementioned methodology, so that we are capable of extracting the predicate dbo:instrument, with $L_{dbo:instrument} > \varepsilon_{max}$, for the sentence in Example 7. For the sentence in Example 8 we observe that $L_{dbo:instrument} \simeq L_{dbo:associatedMusicalArtist}$ with $L_{max} = max(L_{dbo:instrument}, L_{dbo:associatedMusicalArtist})$ and $L_{max} \in (\varepsilon_{min}, \varepsilon_{max})$.

In ***Step 3*** the sentence in Example 8 is sent to the crowdsourcing module for human annotation. It is remarkable, though, that the ⟨ dbr:Elliott_Smith, dbo:instrument, dbr:Guitar ⟩ triple does not exist in DBpedia.

4 Crowdsourcing-Based Relation Extraction

We have already briefly depicted the introduction of crowdsourced annotations in the RDF extraction pipeline in a human-in-the-loop approach in Sect. 1. The following sections will extensively describe the process of selecting the crowdworkers and assigning them appropriate tasks; we will also focus on the answer aggregation step, which will be detailed in Sect. 4.2.

4.1 User Training

The training step of the annotation tool is necessary to direct the user to the type of annotation we require. It should be recalled that the goal of this work is to obtain a dataset that contains alignments such that triples describe only the content of the corresponding sentence and describe it in its entirety. To do this, we have the annotator perform some preliminary training, which is described as follows. Training is performed on sentences belonging to the WebNLG dataset that have been manually reviewed for correct alignments. For each of these sentences, entity and predicate annotations are collected using LamAPI tool. Short descriptions are also collected that can help the annotator choose the appropriate annotation, as in Example 10.

Example 10. Synthpop
Music genre in which the synthesizer is a key instrument

In each sentence, an entity (indifferently subject or object of the sentence) or a predicate is highlighted. The user is asked to choose the appropriate label for the highlighted portion of text from six alternatives, i.e., five possible annotations and the *None of the above* option. Each annotator must work on 15 sentences to complete the training and provide the correct answer for each before going on to the next step in SEALIon. If an answer is selected that is not the correct one, the user will receive one of the following types of feedback: (i) The information conveyed by the sentence does not match with that of the answer you selected; (ii) The concept expressed by the entity in the sentence is more general than the one in the option you selected; (iii) The concept expressed by the option you selected is more general than the one in the entity in the sentence; (iv) There is at least one correct answer provided among the remaining options—when the *None of the above* option is improperly selected.

Such feedback is intended to draw the annotator to generate triples that meet the requirements of completeness and correctness of the concept expressed by the triple w.r.t. the sentence.

4.2 Task Design and Quality Control

In our pipeline, we model micro-tasks, assigned to crowdworkers as single choice questions with at most six options to choose among, that is, the predicates returned by LamAPI described in Sect. 3.2 and a *None* option if no predicate represents the relation.

Unlike the work described in Li et al. [15], which presents the user with a two-option question about whether a sentence expresses a target relation or not, we opted for a wider range of choices; note that when the worker is given multiple options to consider, as in Grosman et al. [12], relation constraints are generally imposed to avoid impossible or meaningless triples. Such constraints become less necessary in our case since LamAPI generally returns predicates capable of expressing a relation between two given entity classes and the crowdsourcing component wanes the chances of very unlikely relations. In a human-in-the-loop

framework like ours, some major questions have to be answered before proceeding to the actual annotation process thus we need to define: (i) **Who are the annotators**: similar studies that involve a human-in-the-loop approach do not detail the workers selection process; usually a small sample of skilled annotators that are already familiar with the domain they are asked to deal with [12,15]. Picking the right intellectual crowd is fundamental to avoiding poor performances and low-quality data at the end of the process, but how can we assess a worker's quality? Modelling a worker requires choosing an accurate strategy to evaluate a worker's performance without relying on self-reports (e.g., questionnaires, interviews), which cannot reflect the actual expertise of the respondent. Instead, a worker probability model, detailed in Sect. 4.2, appears to be much more satisfactory for our purpose; (ii) **Whom to assign a task**: when a task and a group of workers whose qualities are known are defined, we have to identify a subset of our crowd to whom we want to assign that task. Such a problem is commonly known as the Jury Selection Problem [3] and is detailed in Sect. 4.2; (iii) **How to handle contrasting annotations**: answer aggregation is a key step in collecting responses from crowdworkers. Also known as voting strategy, it briefly consists of assigning the same task to multiple workers and aggregate their results; a Weighted Majority Voting strategy is considered in this work, considering each worker's quality and their answer to the task. Our inter-annotator metric will be discussed in Sect. 4.2.

Moreover, to improve the quality of the annotation tasks, we have designed a User Interface that serves to disguise some of the complexity that derives from the tools and processing steps we have employed.

Who are the Annotators When relying on the wisdom of crowds to derive an answer to our tasks, we assume that, even though some diverging participant responses may exist, the overall judgement is still well-grounded. Different background knowledge, expertise or understanding of the task typically result in varying inter-annotator answers that need to be aggregated to resolve the uncertainty. Following the works in [3,29] we model an Individual Error Rate $q_i \in [0, 1]$, corresponding to the probability that the ith participant will give a contrasting answer with regards to the correct task response.

Definition 1. The Individual Error Rate q_i of the ith participant corresponds to the probability that a worker provides a wrong answer

$$q_i = Pr(\text{the worker's vote} — \text{the task's correct answer A}) \tag{1}$$

A ground truth A, which is a binary value 0 (false) or 1 (true), is therefore required to assess the participants' quality; in detail, we employ a subset of the WebNLG dataset described in Sect. 2, for which RDF triples-text alignments are already known, as a Gold Standard and randomly inject it into real tasks. We will only pick simple sentences in order to minimise the chance of misinterpretation. Based on the participant's answers to such golden tasks, we can compute their Individual Error Rate; participants whose $IER \geq 0.9$ are directly discarded.

Unlike training tests [12], submitted before the actual task, the worker is not aware that an evaluation is being performed, thus reducing the chances of biased responses.

Whom to Assign a Task Once the Individual Error Rate for each participant is computed, we need to identify a subgroup of workers capable of showing the best performance throughout the tasks they are presented with. Such a problem is known as the Jury Selection Problem, presented in [3] as:

Definition 2. Given a candidate worker set U with size K, a budget $B \geq 0$, a crowdsourcing model M, the Jury Selection Problem (JSP) consists in selecting a subset $J_y \subseteq U$ with size $1 \leq y \leq K$, so that J_y is allowed according to M and the Jury Error Rate $JER(J_y)$ is minimised.

In SEALIon's pipeline, we assume that there exist a set of workers who, for any altruistic reason (e.g., students and researchers), are capable of performing high-quality annotations without further financial incentives, so that any group of candidates is allowed in J_y, but other models can be considered [3]. Following Definition 2, we compute the Jury Error Rate $JER(J_y)$ of a subset of workers J_y by computing the probability, given J_y, that the number of mistaken workers C, where $0 \leq C \leq y$, is a minority in J_y

$$JER(J_y) = Pr(C \geq \frac{y+1}{2} | J_y) \tag{2}$$

How to Handle Contrasting Annotations When the selected subset of our crowdworkers has completed their tasks, their answers have to be aggregated. The clearest way of drawing a single decision from disparate opinions is Majority Voting [3,15], which outputs the answer supported by at least half of the participants in our selected subset plus one. Given a group of workers J_y, to compute Majority Voting, we need y, which is the size of our subset, to be odd.

Definition 3. Given a voting V_y, a set of binary values representing the response of a crowd J_y with size y, the Majority Voting strategy is defined as follows:

$$MV(J_y) = \begin{cases} 0 & \text{if } \sum j_i \geq \frac{y+1}{2} \\ 1 & \text{if } \sum j_i \leq \frac{y-1}{2} \end{cases} \tag{3}$$

We adopt a more sophisticated metric, the Weighted Majority Voting [17], which also takes into account the Individual Error Rate q_i of each participant in the crowd J_y, computed as in Definition 1, so that more importance is given to votes expressed by highly-reliable workers (i.e. those with low Individual Error Rate).

5 Validation

SEALIon's results have been assessed following similar applications validation processes [4] by leveraging both automatic and manual evaluation. A brief description of common metrics has been provided in the definitions in Sect. 5.1. When addressing the issue of false negatives, we must consider that, as represented in Table 4, our Gold Standard aligns sentences to triples that are not adequately representative of their content and mainly express information inferred from the sentences themselves. The SEALIon process purposely considers these triples to be erroneous because they are not directly related to the meaning of the sentence.

We validate our pipeline on a subset of sentences—disjoint from the Gold Standard described in Sect. 4.2—from the WebNLG+ test set and the T-REx dataset. The sentences have been randomly selected to avoid biased annotations towards a specific domain. The test set counts 120 sentences, equally parted between the two datasets, aligned with 280 triples (180 from the WebNLG dataset and 100 from T-REx) to be used as a reference. The quantity of the sentences was defined in accordance with what has been done in other works in the state of the art [7]. After the automatic annotation, performed as described in Sect. 3.3, we collected 130 triples and sent another 220 predicted triples belonging to 100 sentences to crowdworkers for manual annotation. Each sentence was reviewed at least 45 times by a pool of graduate and undergraduate students in Computer Science related studies and researchers who were given two days to complete the task, including 15 real sentences and 5 Gold Standard sentences to assess Participant Quality. Table 2 summarises statistics about SEALIon performance regarding the WebNLG and T-REx datasets samples; note that SEALIon is capable of extracting a total of 350 triples aligned with 120 sentences, whereas the reference sets only included 280 triples.

5.1 Automatic Evaluation

We calculate Precision P, defined as the number of correct relation annotations over the total number of positive relation annotations, by comparing the results of our pipeline with reference datasets; note that, since our approach has resulted in a larger number of alignments than those in our samples, we need to assume that: (i) Triples that include the same subject and object entities of the reference triple, related by equal or equivalent (e.g., *location* and *cityLocation*) predicates, and are classified as correct by SEALIon are to be considered true positive candidates; (ii) Triples that include the same subject and object entities of the reference triple, related by different predicates (e.g., *knownFor* in SEALIon alignments and *notableWorks* in reference alignments), and are classified as correct by SEALIon are to be considered false-positive candidates; (iii) Triples present in SEALIon alignments but absent from reference datasets are to be omitted at this stage.

The results of the automatic evaluation are presented in Table 3; precision is calculated here before the triple is submitted to the crowdsourcing annota-

tion module and after the sentence has been re-annotated in order to assess the improvement made by using a human-in-the-loop approach. SEALIon performs best on WebNLG sentences even without human annotation, while it does not accomplish high precision on the T-REx dataset. Another precision value has been computed with regards to the system's prediction before human annotations are performed; the automatic relation extraction module highest-scoring predicate (see Sect. 3.3 for a detailed description of the process) has been considered as a prediction of the correct predicate between a pair of entities. Such prediction has been then compared with the crowdsourcing module's outcome, which could confirm or modify the system's prediction. After manual review, 80% of triples are confirmed, reaching an average precision of 0.88. Although an increase in precision resulting from manual annotation is to be expected, it should be emphasised that a minor human effort is sufficient to achieve a significant improvement in the quality of the final output.

The results presented in Table 3 will be better discussed in Sect. 5.2.

Table 2. Statistics of SEALIon's results for each dataset's test sample

	WebNLG	T-REx	Total
Total extracted triples	140	210	350
Triples to be reviewed by crowdworkers	50	170	220
Triples automatically accepted	90	40	130
Average triples per sentence	2	4	3

Table 3. Precision Calculated with Regards to Reference Triples in WebNLG and T-REx

	WebNLG sample		T-REx sample	
	No crowd annotations	With crowd annotations	No crowd annotations	With crowd annotations
Precision P	0.84	0.92	0.40	0.80

5.2 Manual Evaluation

A sample of SEALIon's output (50 sentences, 142 triples) has been collected and manually checked for correct relations by a small group of domain experts, each of which was asked to accept an alignment as correct if and only if the triple was explicitly mentioned in the sentence. Note that, as stated in the previous section, SEALIon was capable of extracting and aligning more triples then those present in reference datasets, especially when compared to T-REx. An example of alignments proposed to manual reviewers is presented in Table 4. Such triple-sentence couples' quality could not be assessed with automatic evaluation since

Table 4. Triple-text alignments in SEALIon compared to the same alignments in reference datasets' samples

The population of the metropolitan area of Ciudad Ayala, a part of Morelos, Mexico, located at 1147 above sea level, is 1 777 539, and the UTC offset for this Pacific Daylight Time zone area is −6.	
SEALIon	**WebNLG**
Ciudad Ayala — part — Morelos	C.A. — populationMetro — 1777539
C.A. — populationMetro — 1777539	Ciudad Ayala — utcOffset — −6
Ciudad Ayala — utcOffset — −6	Ciudad Ayala — isPartOf — Morelos
C.A. — minimumElevation — 1147	Ciudad Ayala — country — Mexico
	C.A. — elevationAboveTheSeaLevel — 1147
	C.A. — timeZone — PacificDaylightTime
Saul Swimmer was an American documentary film director and producer best known for the movie The Concert for Bangladesh (1972).	
SEALIon	**T-REx**
The Concert for Bangladesh (film) — director — Saul Swimmer	Saul S. — countryOfCitizenship — American
Saul S. — genre — documentary film	Bangladesh — diplomaticRelation — American

no reference triples could be found in WebNLG and T-REx. Consider the first sentence in Table 4: SEALIon was capable of correctly extracting four triples out of six references, two of which, ⟨CiudadAlaya, timeZone, PacificDaylightTime⟩ and ⟨CiudadAlaya, country, Mexico⟩ are not directly mentioned in the text and are rather inferred from its content. A similar case is represented by the second sentence, where the ⟨Bangladesh, diplomaticRelation, American⟩ triple is not only presumably inferred from the text, but it is also incapable of accounting for the informational content of the sentence. According to manual reviewers, SEALIon results in an accuracy of 93% in aligning triples and sentences so that the former are relevant to the informational content of the latter.

6 Conclusions and Future Research

In this work, a pipeline has been defined, including an automatic pre-processing and entity recognition module and a crowdsourcing component aimed at collecting a small amount of manually annotated data to be reinserted in the process to reduce the error rate in the extraction of RDF triples. We have then focused on the most critical issues that arise when a crowd of human respondents is considered, from selecting a proper set of annotators to the definition of a metric for computing inter-annotator agreement. SEALIon proves to be a useful tool to perform RDF extraction from natural language and unstructured text. However, more testing needs to be performed to assess its accomplishments outside of controlled textual sources. As depicted, the main goal of the SEALIon solution

is the definition of a methodology for RDF-text alignment. Therefore it is possible to use this methodology to create new datasets that meet the user's needs (e.g., specific domain dataset). We purportedly decided to focus on evaluating a restricted number of alignments, concentrating on testing the improvements provided by the human-in-the-loop approach compared to fully automatic triple extraction pipelines. Still, an analysis of how many epochs are necessary for a single textual source before manual annotation is no longer required and what quality is achieved than has to be carried out. Moreover, further research should be conducted on the interaction between the user and the system in both face-to-face and remote annotation settings to supply sufficient support for high-quality relation extraction in either environment. Of course, different paths could be explored for the techniques and formulas we have presented in this work. It would be interesting to compare state-of-the-art methodologies in a human-in-the-loop approach like ours. Also, long-time testing to verify how much previous manual annotations can impact future automatically extracted triples is foreseeable.

Acronyms

AMT	Amazon Mechanical Turk
EL	Entity Linking
KB	Knowledge Base
KBP	Knowledge Base Population
KG	Knowledge Graph
KGs	Knowledge Graphs
ML	Machine Learning
NEL	Named Entity Linking
NER	Named Entity Recognition
NLG	Natural Language Generation
NLP	Natural Language Processing
QA	Question Answering
RDF	Resource Description Framework
RE	Relation Extraction

References

1. Angeli, G., Tibshirani, J., Wu, J., Manning, C.D.: Combining distant and partial supervision for relation extraction. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1556–1567. Association for Computational Linguistics, Doha, Qatar (2014)

2. Bhattacharjee, B., Kender, J.R., Hill, M., Dube, P., Huo, S., Glass, M.R., Belgodere, B., Pankanti, S., Codella, N., Watson, P.: P2l: Predicting transfer learning for images and semantic relations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 760–761 (2020)
3. Cao, C.C., She, J., Tong, Y., Chen, L.: Whom to ask? jury selection for decision making tasks on micro-blog services. Proc. VLDB Endow. **5**(11), 1495–1506 (2012)
4. Ferreira, T.C., Gardent, C., Ilinykh, N., van der Lee, C., Mille, S., Moussallem, D., Shimorina, A.: The 2020 bilingual, bi-directional WebNLG+ shared task: overview and evaluation results (WebNLG+ 2020). In: Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+), pp. 55–76. Association for Computational Linguistics, Dublin, Ireland (Virtual) (2020)
5. Ferreira, T.C., van der Lee, C., van Miltenburg, E., Krahmer, E.: Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. In: Proceedings of the EMNLP-IJCNLP, pp. 552–562. Association for Computational Linguistics, Hong Kong, China (2019)
6. Cremaschi, M., De Paoli, F., Rula, A., Spahiu, B.: A fully automated approach to a complete semantic table interpretation. Futur. Gener. Comput. Syst. **112**, 478–500 (2020)
7. Elsahar, H., Vougiouklis, P., Remaci, A., Gravier, C., Hare, J., Laforest, F., Simperl, E.: T-REx: A large scale alignment of natural language with knowledge base triples. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), pp. 3448–3452. European Language Resources Association (ELRA), Miyazaki, Japan (2018)
8. Faridani, S., Hartmann, B., Ipeirotis, P.G.: What's the right price? pricing tasks for finishing on time. In: Proceedings of the 11th AAAI Conference on Human Computation, AAAIWS'11-11, pp. 26–31. AAAI Press (2011)
9. Gardent, C., Shimorina, A., Narayan, S., Perez-Beltrachini, L.: Creating training corpora for NLG micro-planners. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 179–188. Association for Computational Linguistics, Vancouver, Canada (2017)
10. Glass, M., Gliozzo, A.: A dataset for web-scale knowledge base population. In: The Semantic Web, pp. 256–271. Springer International Publishing, Cham (2018)
11. Glass, M., Gliozzo, A., Hassanzadeh, O., Mihindukulasooriya, N., Rossiello, G.: Inducing implicit relations from text using distantly supervised deep nets. In: The Semantic Web—ISWC 2018, pp. 38–55. Springer International Publishing, Cham (2018)
12. Grosman, J.S., Furtado, P.H.T., Rodrigues, A.M.B., Schardong, G.G., Barbosa, S.D.J., Lopes, H.C.V.: Eras: improving the quality control in the annotation process for natural language processing tasks. Inf. Syst. **93**, 101553 (2020)
13. Hu, X., Wen, L., Xu, Y., Zhang, C., Yu, P.: SelfORE: Self-supervised relational feature learning for open relation extraction. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 673–3682. Association for Computational Linguistics (2020)
14. Li, G., Wang, J., Zheng, Y., Franklin, M.J.: Crowdsourced data management: A survey. IEEE Trans. Knowl. Data Eng. **28**(9), 2296–2319 (2016)
15. Li, M., Jin, J., Wu, W., Yang, Y., He, L., Yang, J.: A crowdsourcing based human-in-the-loop framework for denoising uus in relation extraction tasks. In: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2019)

16. Lin, X., Li, H., Xin, H., Li, Z., Chen, L.: Kbpearl: a knowledge base population system supported by joint entity and relation linking. Proc. VLDB Endow. **13**(7), 1035–1049 (2020)
17. Littlestone, N., Warmuth, M.K.: The weighted majority algorithm. Inf. Comput. **108**(2), 212–261 (1994)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013). arXiv:1301.3781
19. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, pp. 3111–3119. Curran Associates Inc. (2013)
20. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pp. 1003–1011. Association for Computational Linguistics, Suntec, Singapore (2009)
21. Mrabet, Y., Vougiouklis, P., Kilicoglu, H., Gardent, C., Demner-Fushman, D., Hare, J., Simperl, E.: Aligning texts and knowledge bases with semantic sentence simplification. In: Proceedings of the 2nd International Workshop on Natural Language Generation and the Semantic Web (WebNLG 2016), pp. 29–36. Association for Computational Linguistics, Edinburgh, Scotland (2016)
22. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. CoRR (2019)
23. Riedel, S., Yao, L., McCallum, A., Marlin, B.M.: Relation extraction with matrix factorization and universal schemas. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 74–84. Association for Computational Linguistics, Atlanta, Georgia (2013)
24. Shimorina, A., Khasanova, E., Gardent, C.: Creating a corpus for Russian data-to-text generation using neural machine translation and post-editing. In: Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing, pp. 44–49. Association for Computational Linguistics, Florence, Italy (2019)
25. Simon, E., Guigue, V., Piwowarski, B.: Unsupervised information extraction: Regularizing discriminative approaches with relation distribution losses. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1378–1387. Association for Computational Linguistics, Florence, Italy (2019)
26. Smirnova A., Cudré-Mauroux, P.: Relation extraction using distant supervision: A survey. ACM Comput. Surv. **51**(5) (2018)
27. Yao, L., Riedel, S., McCallum, A.: Collective cross-document relation extraction without labelled data. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 1013–1023. Association for Computational Linguistics, Cambridge, MA (2010)
28. Zhao, C., Walker, M., Chaturvedi, S.: Bridging the structural gap between encoding and decoding for data-to-text generation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 2481–2491. Association for Computational Linguistics (2020)
29. Zheng, Y., Cheng, R., Maniu, S., Mo, L.: On optimality of jury selection in crowdsourcing. In: Proceedings of the 18th International Conference on Extending Database Technology, EDBT 2015, pp. 193–204 (2015)