knowledge

MDPI

Article

Linking Entities from Text to Hundreds of RDF Datasets for Enabling Large Scale Entity Enrichment

Michalis Mountantonakis [1,*] and Yannis Tzitzikas [1,2,*]

[1] Institute of Computer Science, FORTH-ICS, 700 13 Heraklion, Greece
[2] Computer Science Department, University of Crete, 700 13 Heraklion, Greece
* Correspondence: mountant@ics.forth.gr (M.M.); tzitzik@ics.forth.gr (Y.T.)

Abstract: There is a high increase in approaches that receive as input a text and perform named entity recognition (or extraction) for linking the recognized entities of the given text to RDF Knowledge Bases (or datasets). In this way, it is feasible to retrieve more information for these entities, which can be of primary importance for several tasks, e.g., for facilitating manual annotation, hyperlink creation, content enrichment, for improving data veracity and others. However, current approaches link the extracted entities to one or few knowledge bases, therefore, it is not feasible to retrieve the URIs and facts of each recognized entity from multiple datasets and to discover the most relevant datasets for one or more extracted entities. For enabling this functionality, we introduce a research prototype, called $\texttt{LODsyndesis}_{IE}$, which exploits three widely used Named Entity Recognition and Disambiguation tools (i.e., DBpedia Spotlight, WAT and Stanford CoreNLP) for recognizing the entities of a given text. Afterwards, it links these entities to the *LODsyndesis* knowledge base, which offers data enrichment and discovery services for millions of entities over hundreds of RDF datasets. We introduce all the steps of $\texttt{LODsyndesis}_{IE}$, and we provide information on how to exploit its services through its online application and its REST API. Concerning the evaluation, we use three evaluation collections of texts: (i) for comparing the effectiveness of combining different Named Entity Recognition tools, (ii) for measuring the gain in terms of enrichment by linking the extracted entities to *LODsyndesis* instead of using a single or a few RDF datasets and (iii) for evaluating the efficiency of $\texttt{LODsyndesis}_{IE}$.

Keywords: entity recognition; linking; content enrichment; annotation; hyperlink creation; fact checking; RDF; linked open data; data discovery; semantic web

Citation: Mountantonakis, M.; Tzitzikas, Y. Linking Entities from Text to Hundreds of RDF Datasets for Enabling Large Scale Entity Enrichment. *Knowledge* **2022**, *2*, 1–25. https://doi.org/10.3390/knowledge2010001

Academic Editor: Pentti Nieminen

Received: 27 September 2021
Accepted: 28 November 2021
Published: 24 December 2021

1. Introduction

The target of Information Extraction (IE) [1,2] approaches is to extract information either from unstructured (e.g., plain text) or semi-structured (e.g., relational databases) sources. For achieving this objective, a crucial task of an IE process is to perform Entity Recognition (ER) for identifying the entities of the given source (i.e., an entity is a real-world thing that has its own independent existence such as a person, place or a concept) and afterwards to link the recognized entities to an existing knowledge base. Due to the high increase in and popularity of RDF Knowledge Bases (or datasets) [3,4], a high number of ER approaches link the recognized entities of a given source (e.g., text) to popular RDF datasets [5]. In this way, it is feasible to enrich the contents for these entities, which can be of primary importance for several tasks, e.g., for facilitating manual annotation, for enabling hyperlink creation, for offering content enrichment, for improving data veracity and others.

However, a major limitation of existing Entity Recognition tools is that they link each recognized entity (of a given text) mainly to a single knowledge base, e.g., the DBpedia Spotlight tool [6] annotates each entity with a link to DBpedia [7], and the WAT tool [8] provides links to Wikipedia. Thereby, by using such tools, it is not trivial to provide

services for the recognized entities by combining information from multiple datasets, e.g., it is difficult to find all the related URIs (Uniform Resource Identifiers) of each entity, to collect all its triples (i.e., facts) and to verify facts that are included in the given text.

This could be partially achieved by using approaches such as *LODsyndesis* [4,9] and sameAs.cc [10], which provide all the available URIs for an entity from multiple RDF datasets. However, such systems do not perform entity recognition, therefore, the user has to use two or more systems: one ER system and one system that offers entity enrichment by providing information from multiple datasets for a given entity. Another drawback is that the results of different ER tools are not combined. To tackle the above challenges, the hypotheses that we investigate in this article are the following:

Hypothesis 1 (H1). *It is better to combine the merits of different techniques and tools coming from the Natural Language Processing (NLP) and Knowledge Base (KB) community for performing entity recognition (ER).*

Hypothesis 2 (H2). *It is better to link the recognized entities to multiple datasets instead of a single (or a few) ones for providing more advanced services for numerous real-world tasks.*

Based on this objective and the aforementioned hypotheses, we would like to investigate the following: (a) how to combine different ER tools in an effective way and (b) how to link the recognized entities to hundreds of RDF datasets for providing more advanced services for several real world tasks.

For overcoming the above challenges, we introduce a research prototype, called $\texttt{LODsyndesis}_{IE}$ (https://demos.isl.ics.forth.gr/LODsyndesisIE/, accessed on 12 November 2021). This prototype combines a tool from the NLP community (Stanford CoreNLP [11]) and two tools from the KB community (DBpedia Spotlight [6] and WAT [8]) for recognizing the entities of a text and provides fast access to several services (including a REST API) for each recognized entity by exploiting hundreds of RDF datasets simultaneously. In particular, $\texttt{LODsyndesis}_{IE}$ exploits the services of *LODsyndesis*, a suite of services (based on dedicated indices) that support cross-dataset reasoning over hundreds of RDF datasets. The current version of *LODsyndesis* contains 2 billion triples and 400 million entities from 400 RDF datasets. The exploitation of *LODsyndesis* services from $\texttt{LODsyndesis}_{IE}$ enables real time interaction and can bring several benefits for the recognized entities to multiple tasks, including Automatic Annotation, Hyperlink Creation, Entity Enrichment, Data Veracity, Data Discovery and Selection and Data Integration.

This article is an extension of a demo paper [12]. In comparison to that paper, the current paper: contains an extended related work section that describes more related approaches, analyzes in more details the steps of $\texttt{LODsyndesis}_{IE}$, provides the algorithm for recognizing the entities of a text by combining three ER tools, describes the offered services that exploit multiple datasets (through *LODsyndesis*) and details how to use its web application and its REST API. In addition, in the current paper, we evaluate $\texttt{LODsyndesis}_{IE}$ in terms of its effectiveness and efficiency. In particular, we use three evaluation collections: (i) for measuring the gain of combining multiple ER tools for the ER process (indicatively, by combining three ER tools, we achieve on average the highest average recall and F1 Score) , (ii) for evaluating the gain of linking the entities to hundreds of RDF datasets and (iii) for evaluating the efficiency of $\texttt{LODsyndesis}_{IE}$.

The rest of this paper is organized as follows: Section 2 introduces the background and the related work. Section 3 introduces the steps and the offered services of the proposed approach. Section 4 reports comparative experimental results concerning the effectiveness and the efficiency of $\texttt{LODsyndesis}_{IE}$, whereas Section 5 concludes the paper and discusses possible directions for future work.

2. Background and Related Work

In this section, in Section 2.1 we provide details about the RDF data model, whereas in Section 2.2 we introduce related approaches and the novelty of $\texttt{LODsyndesis}_{IE}$.

2.1. Background Resource Description Framework (RDF)

RDF is a graph-based data model. In particular, "an RDF graph (or knowledge base) is a set of subject-predicate-object triples, where the elements may be Internationalized Resource Identifiers (IRIs), blank nodes, or datatyped literals" [13] (see also https://www.w3.org/TR/rdf11-concepts/, accessed on on 12 November 2021). Hereafter, we shall use the term URIs (Uniform Resource Identifiers) to refer to IRIs (since the term URI is more commonly used). A triple is a any element of $T = (U \cup B_n) \times (U) \times (U \cup Bn \cup L)$, where U, Bn and L denote the sets of URIs, blank nodes and literals, respectively. Moreover, an RDF dataset (or Knowledge Base) is any finite subset of T. For example, the triple ⟨dbp:Elijah_Wood, dbp:occupation, dbp:Actor⟩, contains three URIs, where dbp:Elijah_Wood is the subject, dbp:occupation is the predicate (or property) and dbp:Actor is the object (i.e., the value of the predicate). By using the mentioned model, the linking between different RDF datasets can be achieved by the existence of common URIs or Literals or by defining equivalence relationships through ontologies such as OWL [14], e.g., owl:sameAs, among different URIs (e.g., entities). For example, the triple ⟨dbp:Elijah_Wood, owl:sameAs, yago:Elijah_Wood⟩ denotes that the two URIs, dbp:Elijah_Wood and yago:Elijah_Wood, refer to the same real person.

2.2. Related Work

First, we provide information about approaches that perform Entity Recognition, Linking and Annotation over knowledge bases (in Section 2.2.1). Second, we mention approaches that can be used for enriching the content of one or more entities (in Section 2.2.2). Finally, we discuss the novelty of $\texttt{LODsyndesis}_{IE}$ compared to the related approaches (in Section 2.2.3).

2.2.1. Entity Recognition, Linking and Annotation over Knowledge Bases

Here, we mention some popular Entity Recognition tools that link the recognized entities to knowledge bases. Figure 1 depicts the key dimensions that are related to the problem that we focus on in this paper. Concerning the *Input*, such tools can receive as input either unstructured sources (e.g., plain text) or semi-structured sources (e.g., tables of relational databases). Regarding the *Entity Recognition and Disambiguation process*, they can use either pure Natural Language Processing (NLP) methods, methods based on Knowledge Bases (KBs) or methods based on Neural Networks (NN), e.g., word embeddings. Concerning *Entity Linking*, they usually associate each recognized entity with links (URIs) to a knowledge base (e.g., to an RDF dataset), i.e., for reasons of disambiguation and/or for extracting more information from the corresponding KB. The *Output* of such processes is usually the annotation of entities in the text and the creation of hyperlinks to a knowledge base. However, the results can also be exploited for creating *Services* for several other tasks, e.g., for enriching the contents of recognized entities, for Data Veracity, for Question Answering and others.

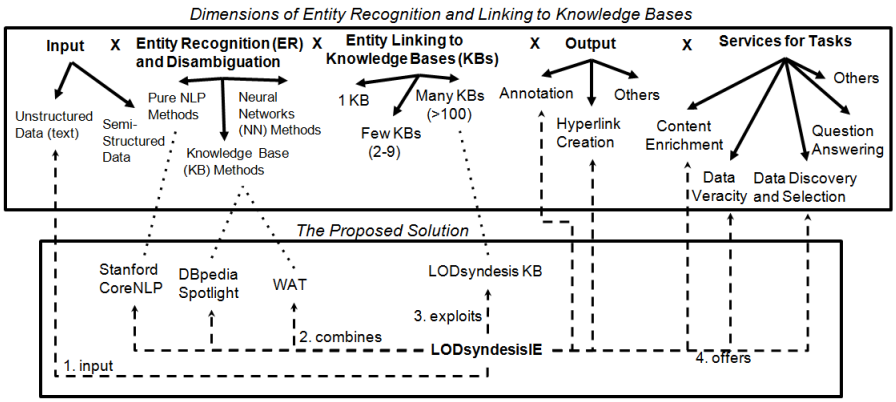

Figure 1. Dimensions of entity recognition and linking to knowledge bases.

Comparison of existing ER tools. Table 1 compares some popular ER tools that link the recognized entities to popular knowledge bases by using the dimensions of Figure 1. These tools usually receive as input a plain text, and regardless of the method that they use for the process of entity recognition and disambiguation, they annotate and link the recognized entities to a single knowledge base, i.e., either to Wikipedia, or to popular RDF datasets, such as DBpedia [7] and YAGO [15]. Moreover, some of these tools provide more services, e.g., Stanford CoreNLP [11] offers part-of-speech tagging and WAT [8] offers entity relatedness services, i.e., for measuring whether two recognized entities are semantically related to each other.

Table 1. Popular ER tools that link the recognized entities to Knowledge Bases.

Tool	Input	ER Method	Linking to KBs	Main Output
Stanford CoreNLP [11]	Text	pure NLP	1 KB (Wikipedia, by using a dictionary)	Part-of-Speech Tagging, Annotation, Hyperlink Creation, others
DBpedia Spotlight [6]	Text	KB	1 KB (DBpedia)	Annotation, Hyperlink Creation
WAT [8]	Text	KB	1 KB (Wikipedia)	Annotation, Hyperlink Creation, Entity Relatedness
Babelfy [16]	Text	KB	1 KB (DBpedia)	Annotation, Hyperlink Creation
AIDA [17]	Text, Tables	KB	1 KB (YAGO)	Annotation, Hyperlink Creation
REL [18]	Text	NN	1 KB (Wikipedia)	Annotation, Hyperlink Creation
SOTA NLP [19]	Text	NN	1 KB (Wikipedia)	Annotation, Hyperlink Creation

Much more details about the techniques of these tools, and many other information extraction approaches over knowledge bases are analyzed in two recent surveys [20,21]. Moreover, Ref. [5] provides a more detailed comparison of ER approaches through the benchmark *GERBIL*, whereas [22] describes the strengths and weaknesses of such tools.

A key observation from both Table 1 and the mentioned papers is that most of the tools exploit a single knowledge base for linking the recognized entities.

Information Extraction for specific tasks over RDF Datasets. In many cases, the ER tools of Table 1 and information extraction techniques over RDF datasets are exploited from other approaches for improving the execution of several tasks, including Question Answering, Fact Checking and others [20]. Indicatively, the tools *WDAqua* [23], *Aqqu* [24], *SINA* [25] and *LODQA* [26] use information extraction techniques (e.g., including entity recognition and linking) for offering Question Answering over Knowledge bases (more tools are surveyed in [27]). Moreover, the tool ClaimLinker [28] offers fact checking by linking a given text to a knowledge graph containing fact-checked claims, whereas [29] combines a semantic annotator with a named entity recognizer, for improving the performance of entity linking and annotation. [30] introduces the tool *Doc2RDFa*, which annotates Web Documents using RDFa, whereas the authors in [31] perform Entity Linking for improving the task of document ranking. Furthermore, Neckar [32] exploits Wikidata KB [33], for assigning name entity classes to Wikidata items. Finally, the *Falcon* tool [34] extracts entities and relations in short texts or questions by using DBpedia.

Evaluation collections. For comparing entity recognition and linking tools, GERBIL benchmark [5] provides a high number of evaluation collections, where most of them rely on DBpedia KB. A recent evaluation collection, called $KORE50^{DYWC}$ [35], links the entities of a text to four popular datasets: DBpedia, YAGO, Wikidata and Crunchbase.

2.2.2. Entity Enrichment over Multiple RDF Datasets

There are several tools and web applications that provide services by exploiting multiple RDF datasets. Specifically, by using services such as WIMU [36], *LODsyndesis* [4,37] and LODLaundromat [38], one can find all the RDF datasets containing a given URI, whereas, through sameAs.cc [10] and *LODsyndesis*, one can find all the available URIs of a given entity, i.e., these approaches compute the transitive and symmetric closure of owl:sameAs relationships. Regarding the triples of each entity, by using LOD-a-LOT [39], one can have access to 28 billion triples from 650K RDF documents from a single self-indexed file, whereas *LODsyndesis* offers more than 2 billion triples from 400 RDF datasets for millions of entities. Another option for enriching the content of a recognized entity is by sending SPARQL (https://www.w3.org/TR/sparql11-overview/, accessed on 12 November 2021) queries to the corresponding endpoints, e.g., to DBpedia [7] or Wikidata [33] SPARQL endpoints, whereas for retrieving data from multiple datasets, one option is to send federated queries [40] among several SPARQL endpoints. All the above approaches can be used for enriching the contents of an entity, however, none of these approaches perform entity recognition.

2.2.3. Placement and Novelty of $\texttt{LODsyndesis}_{IE}$

Comparing to the aforementioned approaches, we do not focus on proposing a new *Entity Extraction* system (e.g., [6,8,11]) or a new service for offering data enrichment (e.g., [9,10,39]). On the contrary, the proposed approach aims at connecting these "two worlds" (see the lower side of Figure 1). In particular, $\texttt{LODsyndesis}_{IE}$ combines existing *Entity Recognition* tools (i.e., DBpedia Spotlight, Stanford CoreNLP and WAT) for recognizing the entities of a given text and uses the *LODsyndesis* knowledge base [37] for linking the recognized entities to hundreds of RDF datasets. Through that process, it can offer more advanced services for the recognized entities (see the lower right side of Figure 1) by combining information from multiple datasets, as it is explained in Section 3.

Concerning the novelty of $\texttt{LODsyndesis}_{IE}$, to the best of our knowledge, it is the first approach that links the recognized entities to hundreds of RDF datasets for offering more advanced services for the entities of any given text.

3. The Proposed Approach of $\texttt{LODsyndesis}_{IE}$

In this section, we first present the steps for recognizing the entities of a given text (in Section 3.1), then we describe how to link them to *LODsyndesis* (in Section 3.2), and finally in Section 3.3, we introduce the offered services for the recognized entities. Figure 2 introduces our running example, where the input from the user is a small text about "The Lord of the Rings Film Series".

3.1. Input and the Entity Recognition Process

By using $\texttt{LODsyndesis}_{IE}$, the user types a text in English language and selects the combination of Entity Recognition (ER) tools that will be used, i.e., the available tools are DBpedia Spotlight (DBS), Stanford CoreNLP (SCNLP) and WAT. In the running example of Figure 2, we chose to use all the three tools. Algorithm 1 shows the exact steps for recognizing the entities of a text by combining all the tools (i.e., our target is to test the Hypothesis H1). However, it can be easily adjusted for using either one or two ER tools.

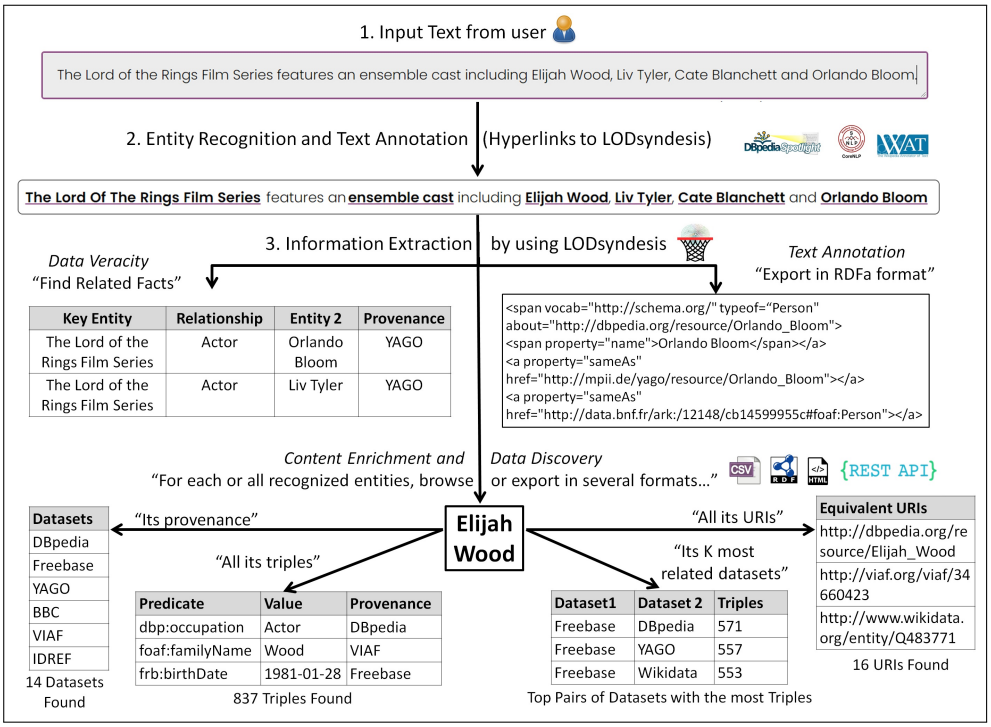

Figure 2. Running example for a text for "The Lord of the Rings Film Series".

3.1.1. Part A. Entity Recognition from Each ER Tool Separately

In the first part of Algorithm 1, we use each ER tool separately for recognizing the entities of the text. Concerning DBpedia Spotlight and WAT (lines 1–2), both ER tools produce a set of entity–URI pairs, i.e., for each entity DBpedia Spotlight provides its DBpedia URI and WAT its Wikipedia URI. For being able to compare the URIs derived from all the three tools, for the WAT tool we replace each Wikipedia URI with its equivalent DBpedia URI. On the other hand, Stanford CoreNLP (lines 4–8) produces just a unique word/phrase for each entity (and not a URI). For this reason, we use the *kwd_to_URIs* service from *LODsyndesis* to find the most relevant DBpedia URI for each recognized entity. In particular, that service takes as input a specific word/phrase e, i.e., a word referring to an entity, and produces a list of URIs whose suffix (i.e., the last part of the URI) starts with the word e , e.g., the suffix of the URI "http://dbpedia.org/resource/Orlando_Bloom, accessed on 12 November 2021," is "Orlando Bloom". From the aforementioned list, we select the URI whose suffix ($suf(u)$) has the minimum Levenhstein distance with e, and we store the selected entity-URI pair to the set EU^{nlp} (lines 7–8).

Example. In the upper side of Figure 3, we can see a possible output for the input text of our running example for each ER tool. In this example, we suppose that DBS identified five entities, whereas SCNLP and WAT recognized four entities. As it can be seen, for some entities, two or more tools assigned different URIs, e.g., for the entity "Orlando Bloom", whereas some entities identified only from a single tool, e.g., the entity "Cate Blanchett", from the WAT tool.

3.1.2. Part B. Combining the Results of ER Tools and Assigning a Single URI to Each Entity

The next step is to assign and store a single URI for each recognized entity. First, our target is to increase the number of recognized entities, and for this reason, we store in a set E the entities that were recognized from at least one tool (line 9). As we will see in Section 4, in this way, the results can be improved predominantly for the recall metric and secondarily for the F1 score. An alternative solution that is worth investigating (as a future work) could be to keep only the entities that are recognized from at least two or from all the ER tools (which could maybe improve the results for the precision metric).

Algorithm 1: The process of *Entities Recognition* by combining three ER tools

Input: A text t given by the user

Output: A set of entity-URI pairs denoted by EU, containing the recognized entities of text t and their corresponding link.

1 Part A:Entity Recognition from each ER tool
2 $EU^{dbs} \leftarrow Spotlight.annotate(t)$
3 $EU^{wat} \leftarrow WAT.annotate(t)$
4 $EU^{nlp} \leftarrow \emptyset$
5 $E^{nlp} \leftarrow SCNLP.annotate(t)$
6 **forall** $e \in E^{nlp}$ **do**
7 $U(e) \leftarrow LODsyndesis.kwd\_to\_URIs(e)$
8 $u_e \leftarrow \arg_u \min\{Levenhstein(suf(u), e) \mid u \in U(e)\}$
9 $EU^{nlp} \leftarrow EU^{nlp} \cup \{(e, u_e)\}$
10 Part B: Combining the results of ER tools and Assigning a single URI to each entity
$E \leftarrow \{\, e \mid (e, u) \in \{EU^{nlp} \cup EU^{dbs} \cup EU^{wat}\}\}$
11 $EU \leftarrow \emptyset$
`/* Iterate over Entities recognized from at least one ER tool */`
12 **forall** $e \in E$ **do**
13 $u_{nlp} = \begin{cases} u & \text{if } (e,u) \in EU^{nlp} \text{(entity e recognized from the tool)} \\ \lambda & \text{if } (e,u) \notin EU^{nlp} \text{(entity e was not recognized from the tool)} \end{cases}$
14 $u_{wat} = \begin{cases} u & \text{if } (e,u) \in EU^{wat} \text{ (entity e recognized from the tool)} \\ \lambda & \text{if } (e,u) \notin EU^{wat} \text{ (entity e was not recognized from the tool)} \end{cases}$
15 $u_{dbs} = \begin{cases} u & \text{if } (e,u) \in EU^{dbs} \text{ (entity e recognized from the tool)} \\ \lambda & \text{if } (e,u) \notin EU^{dbs} \text{ (entity e was not recognized from the tool)} \end{cases}$
16 $u_{sel} \leftarrow \lambda$`// Initialize it as the empty string`
17 **if** $((u_{nlp} \equiv u_{wat} \;||\; u_{nlp} \equiv u_{dbs})\ and\ u_{nlp} \neq \lambda)$ **then**
18 $u_{sel} \leftarrow u_{nlp}$
19 **else if** $(u_{wat} \equiv u_{dbs}\ and\ u_{wat} \neq \lambda)$ **then**
20 $u_{sel} \leftarrow u_{wat}$
21 **else**
22 $u_{sel} \leftarrow \arg_u \min\{Levenhstein(suf(u), e) \mid u \in \{u_{dbs}, u_{nlp}, u_{wat}\}, u \neq \lambda\}$
23 $EU \leftarrow EU \cup \{e, u_{sel})\}$
24 **Return** EU

Afterwards, we iterate over each entity e of set E (line 11), and we store in a single variable the URI of each tool for entity e (lines 12–14). Since we have taken the union of recognized entities, an entity can be recognized either by a single, two or all three ER tools. We suppose that if an entity was not recognized from a specific ER tool, then its corresponding URI equals λ (i.e., the empty string). For instance, in Figure 3, the entity "Liv Tyler" is only recognized from the DBS tool, and not from the other two tools. Therefore, the URI for Liv Tyler for the DBS tool is dbr:Liv_Tyler, whereas the corresponding URI for the other two tools for that entity is λ (i.e., the empty string). For the candidate URIs of the three ER tools, we use the following rules:

Rule A (The Majority Rule): If at least two ER tools recognized the entity e and assigned to e the same URI, then we select this URI for the entity e (see lines 16–19). The lines 16–17 include the cases where either (a) the tools SCNLP and WAT, (b) the tools SCNLP and DBS or (c) all the ER tools, managed to recognize the entity e and assigned to it the same URI. On the contrary, the lines 18–19 are executed when both WAT and DBS recognized e and provided the same URI.

- **Examples of Rule A.** As we can see in the lower part of Figure 3, for the entity "Elijah Wood" all the tools assigned the same URI. On the contrary, for the entity "The Lord of the Rings Film Series", two tools (i.e., DBS and SCNLP) assigned the same URI, whereas WAT assigned a different URI. However, due to the majority rule, we selected the URI returned by DBS and SCNLP.

Rule B (The Closest Name Rule): The lines 20–21 are executed in the following cases: (i) the entity e was recognized only from a single tool, and (ii) the entity e was recognized from two or three tools, however, each of them was assigned a different URI for entity e. In case (i), we have only a single candidate URI, therefore, we select that URI. In case (ii), we have more than one candidate URIs for e, and we select the URI u whose suffix, i.e., the last part of the URI, has the minimum Levenhstein distance with e.

- **Examples of Rule B.** In Figure 3, for the entity "Orlando Bloom", each tool assigned a different URI. However, since the suffixes of the candidate URIs are "Orlando Magic", "Orlando Bloom" and "Orlando, Florida", the algorithm will select the second URI, since its Levenhstein distance with the desired entity is the minimum (i.e., zero). The same case holds also for the entity "Ensemble cast". On the contrary, the entities "Liv Tyler" and "Cate Blanchett" recognized only from a single ER tool, so we selected the URI retrieved from the corresponding tool.

Finally, for each entity, Algorithm 1 stores the selected URI, and in the end, it returns the constructed set of entity–URI pairs, i.e., the set denoted by EU (line 23).

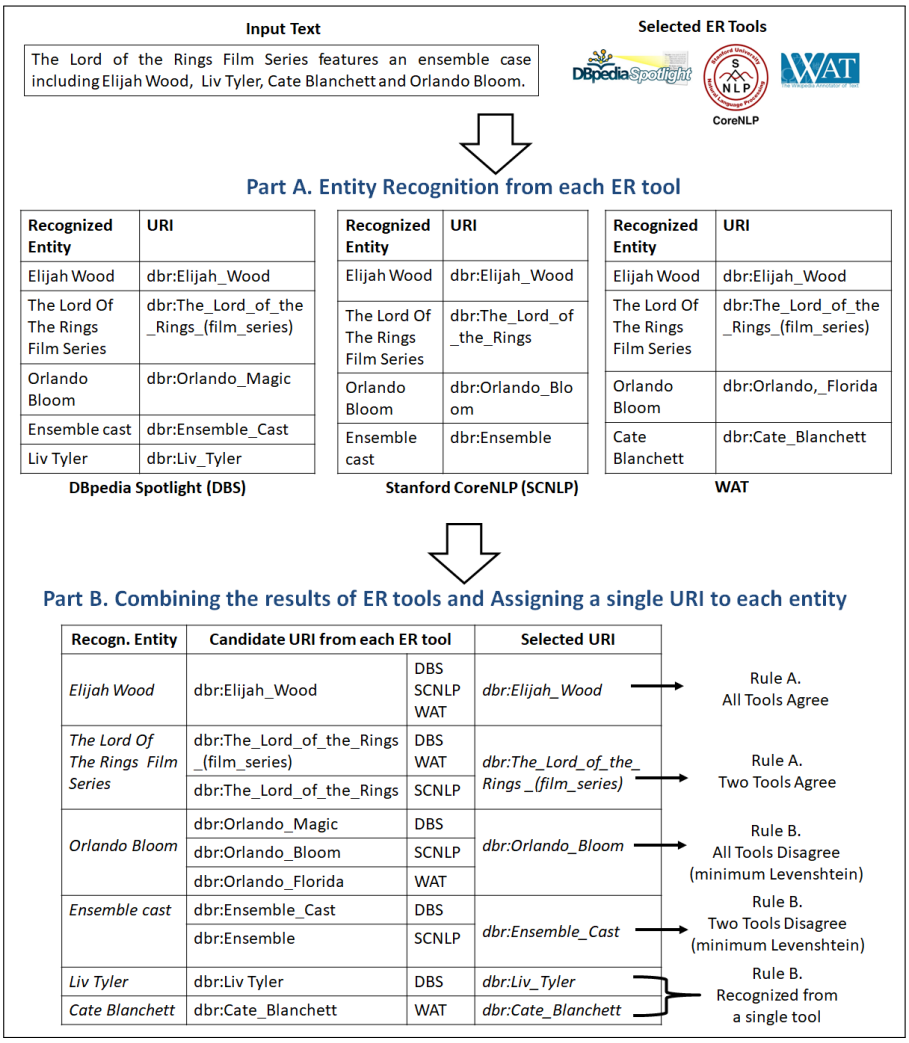

Figure 3. Example of the output of Algorithm 1 for the text of our running example.

3.1.3. Time Complexity

For the first part of the algorithm (lines 5–8), we need to iterate over all the entities recognized from SCNLP, for assigning them a URI. For each entity e, we iterate over the set of candidate URIs, i.e., $U(e)$, which is returned from the *kwd_to_URI* service of *LODsyndesis* whereas $E^{nlp} \subseteq E$, i.e., the set E contains at least the entities recognized from SCNLP. Therefore, in the worst case, we have $|E| * |U(e)_{avg}|$ iterations, where $|U(e)_{avg}|$ is the average number of URIs returned by the *kwd_to_URIs* service for each entity e. Concerning the second part (lines 9–22), we iterate over $|E|$ entities (the set of identified entities from all the tools) for selecting the final URI.

Therefore, the time complexity of Algorithm 1 is $O(|E| * |U(e)_{avg}|)$. Consequently, this algorithm is expected to run fast, especially for small texts where $|E|$ is typically low, whereas $U(e)_{avg}$ is low and configurable.

3.2. Linking the Recognizing Entities to LODsyndesis

The next step is to use the entity–URI pairs of Algorithm 1, for connecting each entity to hundreds of datasets indexed by *LODsyndesis* through hyperlinks (i.e., see Hypothesis H2). For achieving this target, we replace the selected DBpedia URI of each recognized entity with a hyperlink to the corresponding page of the entity to *LODsyndesis* (which contains data from hundreds of RDF datasets), for making it feasible to offer advanced services for the recognized entities (see Section 3.3).

3.3. The Offered Services of $\texttt{LODsyndesis}_{IE}$ *and Its REST API*

Here, we introduce all the services of $\texttt{LODsyndesis}_{IE}$ and the corresponding REST API. $\texttt{LODsyndesis}_{IE}$ exploits the indices and services of *LODsyndesis*for offering more advanced services, either through its web interface or by using its REST API. Concerning the web interface, the user can see the annotated text (see the second step in Figure 2) and an HTML table containing the provided services for each recognized entity in a single table row. In the web application, we also retrieve and show an image for each recognized entity, when it is available. Either by using the web interface or the REST API (see Table 2), we offer services for data annotation and hyperlink creation (see Section 3.3.1), for content enrichment (see Section 3.3.2), for data veracity (see Section 3.3.3) and for data discovery and integration (see Section 3.3.4). Finally, we provide details for the online material of $\texttt{LODsyndesis}_{IE}$ (see Section 3.3.5).

3.3.1. Services for Data Annotation and Hyperlink Creation

$\texttt{LODsyndesis}_{IE}$ offers services for exporting the annotated text either in simple HTML format or in HTML+RDFa format. In particular, each entity is annotated to the output file with its *LODsyndesis* and DBpedia URIs, its type (e.g., "Person") and all its available URIs (derived by *LODsyndesis*) by using the *schema.org* vocabulary [41].

Example. In the middle right side of Figure 2, we can see an example of exporting the annotated text in HTML+RDFa format.

How to exploit these services. Through the web application, one can see the text with the hyperlinks to *LODsyndesis* in HTML format or to download the text in HTML + RDFa format by clicking on the corresponding button. Concerning the REST API, one can use the Service 1 of Table 2 for retrieving the annotated text in HTML+RDFa format. Moreover, one can use the Service 2 of Table 2 (without the optional parameters) for retrieving all the recognized entities, their corresponding DBpedia and *LODsyndesis* URIs and their provenance in tsv (tab separated values) or Ntriples formats.

Table 2. LODsyndesisIE REST API—GET Requests.

ID	Service Name	Description	Parameters	Response Types
1	exportAsRDFa	Recognizing all the entities of a text and annotating the entities by using HTML+RDFa format.	**text**: A text of any length. **ERtools**: one of the following: [WAT, SCNLP, DBS, DBSWAT, SCNLPWAT, DBSCNLP, ALL].	text/html
2	getEntities	Recognizing all the entities of a text and returns for each entity its DBpedia URI, its LODsyndesis URI, and optionally all its available URIs and its provenance.	**text**: A text of any length. **ERtools**: one of the following: [WAT, SCNLP, DBS, DBSWAT, SCNLPWAT, DBSCNLP, ALL] **equivalentURIs**: True for returning all the available URIs for each recognized entity (optional). **provenance**: True for returning all the available datasets for each recognized entity (optional).	text/tsv, application/ n-triples
3	getTriples OfEntities	Recognizing all the entities of a text and producing all the triples of each recognized entity from 400 RDF datasets.	**text**: A text of any length. **ERtools**: one of the following: [WAT, SCNLP, DBS, DBSWAT, SCNLPWAT, DBSCNLP, ALL]	application/ n-quads
4	findRelated Facts	Recognizing all the entities of a text and produces all the facts between the key entity of the text and any other entity. The key entity is by default the first entity of the text, but the user can optionally give an other entity.	**text**: A text of any length. **ERtools**: one of the following: [WAT, SCNLP, DBS, DBSWAT, SCNLPWAT, DBSCNLP, ALL]. **keyEntity**: Put a URI of an entity (optional).	application/ n-quads
5	textEntities DatasetDiscovery	Recognizing all the entities of a text. For the recognized entities, it returns the K datasets a) whose union contains the most triples for these entities (coverage) or b) that contains the most common triples for these entities (commonalities).	**text**: A text of any length. **ERtools**: one of the following: [WAT, SCNLP, DBS, DBSWAT, SCNLPWAT, DBSCNLP, ALL] **resultsNumber**: Number of Results. It can be any integer greater than 0, default is 10 (optional). **subsetK**: Number of Datasets: [1,2,3,4,5], default is 3 (optional). **measurementType**: It can be one of the following: [coverage,commonalities], default is coverage (optional).	text/csv

3.3.2. Services for Content Enrichment

`LODsyndesis`$_{IE}$ offers content enrichment services for browsing or exporting for each entity for all (or a subset of) its available datasets, URIs and triples (or facts). Therefore, the user has the ability to select the URI (and its provenance) that is more desirable for a given task (i.e., this process is also related to the task of Dataset Discovery), or the URI that corresponds to the desired meaning of the word occurrence. Moreover, the user can find complementary facts for each entity even by datasets from different domains, e.g., for the actor "Elijah Wood", a general cross-domain dataset (such as DBpedia) can contain data about his origin, whereas a dataset from the media domain can contain links to his interviews.

Example. In Figure 2, we can see in the lower side a real example for the entity "Elijah Wood", i.e., through `LODsyndesis`$_{IE}$ one can discover which are the datasets containing data about that entity (i.e., 14 datasets) and to retrieve all its available URIs (i.e., 16 URIs) and its triples (i.e., in total 837 triples from 14 datasets).

How to exploit these services. Through the web application, one can download for each single entity (or for all the entities) all its available URIs, datasets and triples in RDF and JSON formats. Moreover, one can browse this information through the web page of *LODsyndesis*. Regarding the Rest API, one can use the Service 2 of Table 2 for retrieving the URIs and datasets for all the recognized entities in tsv and Ntriples format (by using the two optional parameters), and the Service 3 for downloading the triples of all the recognized entities.

For retrieving the desired data for a single entity (and not for all the recognized entities of the text) through a REST Get Request, one can use the REST API of *LODsyndesis*. More details are given in Chapter 6 of [4], whereas the services of *LODsyndesis* are available in https://demos.isl.ics.forth.gr/lodsyndesis/, accessed on 12 November 2021.

3.3.3. Services for Data Veracity

$\texttt{LODsyndesis}_{IE}$ offers a service for browsing or exporting all the relationships between any pair of recognized entities, i.e., all the triples connecting a pair of entities. Moreover, since *LODsyndesis* stores the provenance of each fact, it is possible to check whether a fact is confirmed from a single dataset or from multiple datasets.

Example. In the middle right side of Figure 2, we can see that through $\texttt{LODsyndesis}_{IE}$ we found that the fact "Orlando Bloom acted in The Lord of the Rings Film series" is verified by the YAGO dataset.

How to exploit this service. One can either use the web application or the Service 4 (see Table 2) of the REST API. This service finds facts existing in *LODsyndesis* between a "key entity" and any other possible entity in the text. In the default case, the "key entity" is the first recognized entity of the text. For instance, in Figure 2, by default the key entity is "The Lord of the Rings Film Series", therefore, this service will return facts between that entity and any other entity in the text. However, the user can optionally select any other entity as the "key entity". The results can be exported in N-Quads format.

3.3.4. Services for Dataset Discovery and Selection and for Data Integration

$\texttt{LODsyndesis}_{IE}$ offers services for discovering and selecting the most relevant datasets for one or more entities. The major problem is that, in many cases, the number of available datasets for one or more entities is high, e.g., for the entity "Elijah Wood", there are 14 datasets in *LODsyndesis*. However, sometimes the user desires to select and integrate (either by using a mediator or a semantic warehouse) only K datasets (say $K = 5$) for a set of entities (e.g., for the entities of the given text), since it can be very expensive to integrate several datasets, especially as the number of datasets increases [42]. Through $\texttt{LODsyndesis}_{IE}$ one can find answer to queries of the form "Give me the K datasets that maximize the information (i.e., triples) for an entity (or a set of entities) of the given text", i.e., "the union of these K datasets contains the maximum number of triples for the given entities, comparing to any other combination of K datasets." The process of answering such queries, which includes the execution of content-based measurements among any subset of datasets, is described in [4,37].

Example. In the lower right side of Figure 2, we can see that the pair of datasets offering the most triples for "Elijah Wood" is DBpedia and Freebase with 571 triples.

How to exploit this service. By using the web interface, one can browse (or download in csv format) the K datasets whose union contains the most triples for one or more entities of the text, whereas one can use the Service 5 of Table 2 for retrieving the csv file programmatically through a REST Get Request. The user can optionally select the number of results, the size of K, e.g., for $K = 4$ it will return combinations of four datasets. Finally, it also supports other measurement types that are based on commonalities measurements (more details are given in [4]).

3.3.5. Online Material of LODsyndesis$_{IE}$

The web application and more information about the REST API can be accessed through https://demos.isl.ics.forth.gr/LODsyndesisIE/ (accessed on 12 November 2021), whereas there is an online tutorial video available (https://youtu.be/i52hY57dRms, accessed on 12 November 2021). Moreover, a REST java client and guidelines for using it can be accessed through the following github link: https://github.com/SemanticAccessAndRetrieval/LODsyndesisIE_Java_Client (accessed on 12 November 2021).

4. Evaluation

Here, we evaluate the gain of using multiple ER tools and of linking the entities to hundreds of RDF datasets. In particular, we use three evaluation collections (see Section 4.1) and several metrics (see Section 4.2) for evaluating the effectiveness of LODsyndesis$_{IE}$, i.e., for testing the Hypotheses H1 and H2. Regarding H1, in Section 4.3, we evaluate the gain of combining different tools for Entity Recognition, whereas in Section 4.4, we report measurements related to H2, i.e., for evaluating the gain of using multiple datasets for the recognized entities for several tasks. Finally, Section 4.5 reports efficiency results, and Section 4.6 discusses the results.

4.1. Evaluation Collections

There are several collections for evaluating the task of entity recognition, e.g., the paper of the GERBIL benchmark lists 25 collections [5]. Most of these collections, i.e., 20 out of 25, contain on average five or less entities per text (or document) and usually less than 30 words. Moreover, they usually contain words that are difficult to disambiguate. Our major target in this paper is to use texts with multiple words and entities that cover different domains for evaluating (a) the gain of combining different ER tools for the ER process, (b) the gain in terms of enrichment, by linking the recognized entities to multiple RDF datasets, and (c) the efficiency. We provide experiments by using both simple texts (that do not focus on ambiguous words) and more complex texts.

Specifically, we have created a collection of 10 simple texts covering different domains, called *SimpleWiki*. In addition, for checking the effectiveness of combining different ER tools in larger and more complex texts, we use two existing evaluation collections containing hundreds of words and multiple entities, i.e., *MSNBC* [43] and *AQUAINT* [44]. Below, we describe the three evaluation collections (see Table 3) and we mention the tasks where we use them. All the collections are available online in http://islcatalog.ics.forth.gr/el/dataset/lodsyndesisie-collection (accessed on 12 November 2021). In the same web page, one can also find the exact results for the texts of each collection.

Table 3. The evaluation collections and the tasks where they are used.

Evaluation Collection	Topic	Number of Texts	Entities per Text	Words per Text	Evaluation Task
SimpleWiki	Wikipedia texts	10	15.80	83.2	Effectiveness of ER tools, Data Enrichment, Efficiency.
MSNBC [43]	news	20	37.35	543.9	Effectiveness of ER tools
AQUAINT [44]	news	50	14.54	220.5	Effectiveness of ER tools

SimpleWiki Collection: For creating this evaluation collection, we manually collected 10 texts covering different domains from the Wikipedia Knowledge base. On average (see Table 3), each text contains 15.8 entities and 83.2 words, whereas, in total, all the texts include 158 entities. Table 4 introduces the texts of the SimpleWiki Collection. We can

see for each text a small description (i.e., the main subject of its text), the number of its entities and the number of its words. Moreover, we have also manually created the gold standard. Specifically, we identified each entity in the text and we manually assigned its corresponding DBpedia URI.

MSNBC Collection [43]: It contains 20 texts (see Table 3) , i.e., the top 2 stories in the 10 MSNBC news categories. Each text contains on average 37.35 entities and 543.9 words. We have downloaded the evaluation collection and the gold standard, which includes the Wikipedia URI for each entity, from http://cogcomp.cs.illinois.edu/Data/ACL2011WikificationData.zip (accessed on 12 November 2021). However, we have replaced for each entity the Wikipedia URI with the corresponding DBpedia URI.

AQUAINT Collection [44]: This collection (see Table 3) contains 50 texts that describe news from Xinhua News Service, The New York Times and the Associated Press. It contains on average 14.54 entities and 220.5 words per text. We have downloaded the AQUAINT collection from the same link as MSNBC, and we also replaced the Wikipedia URIs with their corresponding DBpedia URIs (in the gold standard).

The tasks for each collection. We use all the three collections for evaluating the effectiveness of combining different ER tools. On the contrary, we use the SimpleWiki collection for evaluating the gain of data enrichment by linking the recognized entities to multiple RDF datasets and for measuring the efficiency of $\texttt{LODsyndesis}_{IE}$.

Table 4. SimpleWiki Evaluation collection: The description and statistics of the 10 simple texts.

TextID	Text Description	# of Total Words	# of Entities
T1	Lord of the Rings Film Series	97	27
T2	Nikos Kazantzakis (greek writer)	86	13
T3	Scorpions (band)	81	21
T4	The Godfather (Movie)	84	20
T5	2011 NBA Finals	70	12
T6	Argonauts (greek mythology)	80	13
T7	Taj Mahal Mausoleum	67	10
T8	Semantic Web and Tim Berners Lee	92	15
T9	Phaistos Disc (minoan civilization)	75	12
T10	Aristotle (Philosopher)	100	15

4.2. Evaluation Metrics and Hardware Setup

Effectiveness Metrics. Here, we would like to use metrics for testing Hypothesis H1 and H2. Regarding Hypothesis H1, we measure the effectiveness of Entity Recognition (ER) and the linking process. For the recognized entities of the text, there are three different cases with respect to the gold standard:

- **true positives (tp)**: The entities that are recognized from the ER tool(s) and their selected DBpedia URIs are exactly the same as in the gold standard. For instance, suppose that in our running example we use only DBS (i.e., see the table in the upper left part of Figure 3) for recognizing the entities. In that case, "Elijah Wood" is a true positive, i.e., the entity recognized and its URI is correct.
- **false positives (fp)**: The entities are recognized from the ER tool(s), but their selected DBpedia URIs are different from the DBpedia URIs in gold standard. For example, suppose again that we use only DBS in Figure 3. In that case, the entity "Orlando Bloom" is a false positive, i.e., the tool recognized the entity, but the link was erroneous (i.e., "dbr:Orlando Magic"). Moreover, we treat as false positives the cases that are entities found by the ER tools, however, they are not in the gold standard, e.g., suppose in our running example a hypothetical case, where a tool identified the word "Wood" as an entity (and not "Elijah Wood", which is the desired entity).

- **false negatives (fn)**: The entities that are part of the gold standard but failed to be recognized from the ER tool(s). For instance, by using only "DBS" in Figure 3, the entity "Cate Blanchett" is a false negative, since it was not recognized from DBS.

We use the above cases, i.e., tp, fp and fn, for measuring the precision, recall and F1 score as follows:

$$\textbf{Precision} = \frac{tp}{tp+fp} \qquad \textbf{Recall} = \frac{tp}{tp+fn} \qquad \textbf{F1score} = \frac{2*Precision*Recall}{Precision+Recall}$$

As regards Hypothesis H2, we provide some general *LODsyndesis* statistics and we measure the gain of linking the entities of *SimpleWiki* collection to *LODsyndesis* for the tasks of *Content Enrichment*, *Data Veracity*, *Dataset Discovery and Selection* and *Data Integration*.

Efficiency Metrics. We measure the execution time of all the services of $\texttt{LODsyndesis}_{IE}$ (see Services 1–5 of Table 2) by using different combinations of ER tools.

Hardware Setup. For all the experiments, we use a single machine in okeanos cloud computing service (https://okeanos.grnet.gr/, accessed on 12 November 2021) having 8 cores, 8 GB main memory and 60 GB disk space.

4.3. Effectiveness of Combining Different Entity Recognition Tools for ER and Linking (Hypothesis H1)

First, we should mention that we have used the default parameters for each ER tool. In particular, we do not focus on testing several different parameters for each tool (e.g., confidence), i.e., our target is to evaluate whether it is effective to combine different Entity Recognition tools for both (i) simple texts (through SimpleWiki collection) and (ii) more complex texts (by using AQUAINT and MSNBC collections). For each collection, we measured the metrics for each text individually (one can see the exact results for each text online in http://islcatalog.ics.forth.gr/el/dataset/lodsyndesisie-collection, accessed on 12 November 2021), and then we computed the average precision, recall and F1 Score.

Results for SimpleWiki Collection. Table 5 contains the results for the average precision, recall and F1 Score for the SimpleWiki collection. For this collection, by combining more ER tools, the average recall and F1 score always increases. Concerning the performance of each ER tool, by using only DBS, we obtained on average better recall and F1 score versus WAT or SCNLP, whereas by using WAT, we obtained higher precision. However, by using two tools instead of a single one, in all cases the recall and the F1 score highly increased, whereas we obtained the best precision (among any possible combination of tools) by combining DBS and WAT (i.e., 0.911). Finally, by using all three tools, we managed to obtain on average the best recall and F1 scores (i.e., 0.857 and 0.85, respectively).

On the contrary, Table 6 presents for each text the maximum precision, recall and F1 score and the combination of tools that was used in each case. In most cases, we identified the best recall (8 out of 10 cases) and F1 score (6 out of 10 cases) by using all the ER tools. Regarding precision, we can see that in all the cases the most effective combination of tools include either DBS or WAT or both of them.

Table 5. Average precision, recall and F1 score for **SimpleWiki Collection** by using different combinations of recognition tools.

Selected Tool (s)	Avg Precision	Avg Recall	Avg F1 Score
DBpedia Spotlight (DBS)	0.854	0.683	0.752
Stanford CoreNLP (SCNLP)	0.752	0.479	0.568
WAT	0.899	0.607	0.711
DBS+SCNLP	0.833	0.800	0.809
DBS+WAT	**0.911**	0.777	0.831
SCNLP+WAT	0.831	0.699	0.747
All tools (DBS+SCNLP+WAT)	0.857	**0.857**	**0.850**

Table 6. Maximum precision, recall and F1 score for each text of **SimpleWiki** Collection.

Text	Tool & Max Prec.	Tool & Max Rec.	Tool & Max F1 Sc.
(T1) Lord of the Rings	(WAT) 1.000	(All tools) 0.888	(All tools) 0.888
(T2) N. Kazantzakis	(DBS + WAT) 0.923	(+ WAT) 0.923	(DBS + WAT) 0.923
(T3) Scorpions Band	(All tools) 0.900	(All tools) 0.857	(All tools) 0.878
(T4) Godfather	(All tools) 1.000	(All tools) 0.900	(All tools) 0.947
(T5) 2011 NBA Finals	(DBS) 0.750	(DBS) 0.818	(DBS) 0.782
(T6) Argonauts	(WAT) 1.000	(All tools) 1.000	(DBS + WAT) 0.923
(T7) Taj Mahal	(DBS + WAT) 1.000	(All tools) 0.900	(All tools) 0.857
(T8) Semantic Web	(DBS + WAT) 1.000	(All tools) 1.000	(All tools) 0.967
(T9) Phaistos Disc	(All tools) 1.000	(All tools) 0.583	(All tools) 0.736
(T10) Aristotle	(WAT) 0.900	(All tools) 0.800	(DBS + WAT) 0.785

Results for MSNBC Collection. Table 7 shows the results for the MSNBC collection. In this case, we obtained lower precision, recall and F1 score, since it contains more complex texts in comparison to the SimpleWiki collection. Similarly to SimpleWiki, the combination of all tools provided the maximum average recall, whereas the combination of the WAT and DBS tools also achieved a high recall. On the contrary, for that collection, the maximum average precision and F1 score obtained by using only the WAT tool, which was more effective compared to the other two ER tools.

Table 7. Average precision, recall and F1 score for **MSNBC Collection** by using different combinations of recognition tools.

Selected Tool(s)	Avg Precision	Avg Recall	Avg F1 Score
DBpedia Spotlight (DBS)	0.378	0.434	0.387
Stanford CoreNLP (SCNLP)	0.449	0.304	0.376
WAT	**0.544**	0.487	**0.512**
DBS + SCNLP	0.314	0.433	0.356
DBS + WAT	0.350	0.504	0.407
SCNLP + WAT	0.477	0.465	0.469
All tools (DBS + SCNLP + WAT)	0.346	**0.509**	0.405

Results for AQUAINT Collection. Table 8 shows the results for the *AQUAINT* collection. Similarly to the MSNBC collection, we obtained lower values compared to the *SimpleWiki* collection. However, again, we managed to obtain the maximum average recall by combining all the ER tools. It is worth mentioning that we obtained a high increase (+0.095) in the recall by combining all the ER tools instead of using the single ER tool with the highest recall (i.e., DBS). On the contrary, the tool SCNLP was the most effective regarding the precision (however, its recall was too low). Finally, concerning the F1 score, the combination of WAT and DBS was the most effective.

Table 8. Average precision, recall and F1 score for **AQUAINT Collection** by using different combinations of Recognition tools.

Selected Tool (s)	Avg Precision	Avg Recall	Avg F1 Score
DBpedia Spotlight (DBS)	0.463	0.401	0.421
Stanford CoreNLP (SCNLP)	**0.579**	0.242	0.326
WAT	0.555	0.381	0.435
DBS + SCNLP	0.450	0.434	0.434
DBS + WAT	0.465	0.493	**0.472**
SCNLP + WAT	0.514	0.375	0.419
All tools (DBS + SCNLP + WAT)	0.461	**0.496**	0.470

Results by combining the texts of All Evaluation Collections. Table 9 presents the results by combining all the 80 texts of the three evaluation collections. As we can see, the combination of all ER tools achieved the highest average recall and F1 score, whereas the combination of DBS and WAT was quite effective, too. On the contrary, the tool WAT offered the highest average precision, however, its recall value was lower in comparison to combine all the 3 ER tools.

More Results. In Appendix A, we introduce the box plots for each of the three evaluation collections, for the precision, the recall and the F1 Score.

Table 9. Average precision, recall and F1 score by combining the results of the texts of all collections (**SimpleWiki, AQUAINT and MSNBC Collection**) by using different combinations of recognition tools.

Selected Tool (s)	Avg Precision	Avg Recall	Avg F1 Score
DBpedia Spotlight (DBS)	0.491	0.444	0.454
Stanford CoreNLP (SCNLP)	0.568	0.287	0.369
WAT	**0.595**	0.436	0.489
DBS + SCNLP	0.464	0.479	0.462
DBS + WAT	0.492	0.532	0.500
SCNLP + WAT	0.544	0.438	0.472
All tools (DBS + SCNLP + WAT)	0.482	**0.544**	**0.501**

4.4. The Benefits of Linking the Entities to Multiple RDF Datasets (Hypothesis H2)

Here, we measure the gain of linking the recognized entities to multiple RDF datasets for the tasks of Content Enrichment, Data Veracity, Dataset Discovery and Selection and Data Integration by using the SimpleWiki collection.

4.4.1. Measurements for Content Enrichment and Data Veracity

Here, we provide some general statistics from *LODsyndesis* for showing the gain of linking the entities to *LODsyndesis*. Indicatively, we can see in Table 10 that over 25 million entities can be found in two or more datasets indexed by *LODsyndesis*. Moreover, there exist 9075 pairs of datasets sharing at least one entity, whereas 14 million facts can be verified from two or more datasets. As regards the 158 entities of SimpleWiki collection, in Table 11, we provide the number of average URIs, datasets and triples for each entity by using either a single RDF dataset (DBpedia or Wikidata), two RDF datasets (DBpedia and Wikidata) or all the available datasets of *LODsyndesis*. Indicatively, each entity (of SimpleWiki collection) exists on average in 8.5 datasets, and there are 13.8 URIs available for each entity. Concerning the triples, by linking the entities to multiple datasets through *LODsyndesis*, we have access to 208.7% triples per entity versus using only DBpedia and

a 284.1% increase versus using only Wikidata. Finally, regarding Data veracity, by using *LODsyndesis*, we were able to verify 411.1% more facts (i.e., facts that occur in the texts of SimpleWiki collection) compared to using only DBpedia and Wikidata.

Table 10. *LODsyndesis* General statistics for common Entities and Facts.

Measurements for Entities and Facts	Value
# of Entities in ≥2 Datasets	25,289,605
# of Entities in ≥3 Datasets	6,979,109
# of Entities in ≥5 Datasets	1,673,697
# of Entities in ≥10 Datasets	119,231
Pairs of Datasets having at least 1 common entity	9075
# of verifiable facts from ≥2 Datasets	14,648,066
# of verifiable facts from ≥3 Datasets	4,348,019
# of verifiable facts from ≥5 Datasets	53,791
# of verifiable facts from ≥10 Datasets	965
Pairs of Datasets having at least 1 common fact	4468

Table 11. Statistics of linking the recognized entities to DBpedia, Wikidata and to multiple datasets through *LODsyndesis* for the SimpleWiki Collection.

Measurement/Datasets	Linking Only to DBpedia	Linking only to Wikidata	Linking to DBpedia, Wikidata	Linking to *LODsyndesis*
avg URIs per recognized entity	1.0	1.0	2.0	13.2
avg Datasets per recogn. entity	1.0	1.0	2.0	8.5
avg Triples per recogn. entity	406.7	326.8	733.5	1255.5
avg Verified Facts per recognized entity from ≥2 Datasets	–	–	1.8	9.2

4.4.2. Measurements for Dataset Discovery and Selection and Integration

Here, we provide a different scenario, i.e., suppose that one desires to discover, select and integrate the top-K datasets (say K = 3) that maximize the number of triples for the recognized entities of each text. In Table 12, the second column shows the number of datasets containing at least one entity of a text, the third column the total distinct triples for all the entities of each text from all the datasets of *LODsyndesis*, the fourth column shows the top three datasets that enrich at most the content for the recognized entities of each text, whereas in the fourth column, we can see the number of triples included in the union of the top three datasets. For instance, for the text T1, there exists 22 datasets containing in total 21,852 triples for the entities of T1. From these 22 datasets, the combination of 3 datasets whose union contains the most triples of these entities (comparing to any other triad of datasets) is DBpedia, Wikidata and Freebase with 17,393 triples (79.5% of total triples). The key point of Table 12 is that there is not a single triad of datasets that is the most relevant for each of the texts of the collection. Generally, for a given text, an RDF dataset can be quite relevant (e.g., a dataset containing data about geography is highly relevant for texts containing information about places), whereas for another text, the same dataset can be irrelevant (e.g., the same geographical dataset is irrelevant for a text describing animals). However, $\texttt{LODsyndesis}_{IE}$ can detect the most suitable datasets for each such case and can measure (through the services provided by *LODsyndesis*) how relevant a dataset is for the entities of a given text.

Table 12. Dataset discovery and selection scenario: The top-K datasets for enriching at most the content of the entities of each text of SimpleWiki collection.

Text	# of Total Data Sets	# of Total Triples	Top 3 Datasets than Enriches at Most the Content of the Text Entities	# of Triples (Top 3 Datasets)
(T1) Lord of the Rings	22	21,852	DBpedia, Wikidata, Freebase	17,393
(T2) N. Kazantzakis	18	7035	DBpedia, Wikidata, Freebase	4292
(T3) Scorpions Band	27	66,782	DBpedia, Wikidata, Freebase	64,003
(T4) Godfather Movie	20	16,514	DBpedia, Wikidata, Freebase	12,723
(T5) 2011 NBA Finals	13	7540	DBpedia, YAGO, Freebase	6,186
(T6) Argonauts	18	3607	YAGO, Wikidata, Freebase	2471
(T7) Taj Mahal	15	5350	BNF, Wikidata, Freebase	4284
(T8) Semantic Web	48	35,004	DBpedia, Wikidata, BNF	23,635
(T9) Phaistos Disc	20	6484	DBpedia, Wikidata, BNF	4788
(T10) Aristotle	34	22,837	DBpedia, Wikidata, Freebase	13,421

4.5. Efficiency of LODsyndesis$_{IE}$

Here, we measure the efficiency of LODsyndesis$_{IE}$ by using different combinations of ER tools and the texts of SimpleWiki collection for all the services 1–5 of Table 2. We should note that we measure the worst case, i.e., the case where the user wants to download the desired data (i.e, URIs, triples, provenance) for all the recognized entities of the text (and not for a subset of them).

First, we measure the average execution time (in seconds) for four different services of LODsyndesis$_{IE}$, i.e., services 1, 2, 4 and 5 of Table 2. The results are shown in Figure 4. As we can see for each of these services, less than 7 seconds are needed on average for a given text, even by using all the available ER tools (WAT, DBS and SCNLP).

For each different service of Figure 4, it was faster to use WAT, DBS or a combination of them. On the contrary, all the combinations containing SCNLP were slower. However, even by combining all three ER tools, LODsyndesis$_{IE}$ needed on average 4.3 s for the ER and Annotation process, 5.7 s for ER and for retrieving all the URIs and the provenance of each recognized entity, 6.3 s for ER and for discovering the top-K datasets (we used K = 5) for the recognized entities and 7 s for performing ER and fact checking among the entities of the text.

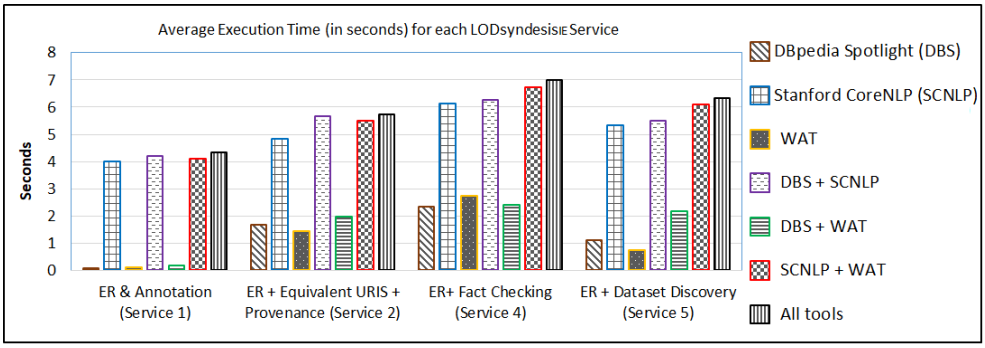

Figure 4. Average execution time (in seconds) for each different LODsyndesis$_{IE}$ service by using any combination of tools for SimpleWiki collection.

On the other hand, it is quite slower to retrieve the triples for all the recognized entities (i.e., Service 3 of Table 2). In particular, we needed on average 542 s for retrieving the triples for the entities of each text by using all the ER tools. Moreover, Figure 5 shows the exact execution time (in minutes) for retrieving the triples for the entities of each text (by using

all the ER tools). Specifically, for eight texts we needed less than 8 min and for three of them less than 1 min. However, for two texts (i.e., Texts T3 and T8), we needed more than 25 min, which is rational since the number of triples were higher for the entities of these texts comparing to the other texts (see the third column of Table 12). However, through the services offered by LODsyndesis$_{IE}$ and *LODsyndesis*, it is feasible to download the triples for a single or a few entities of each text, which can be very fast, especially for entities having a low number of triples.

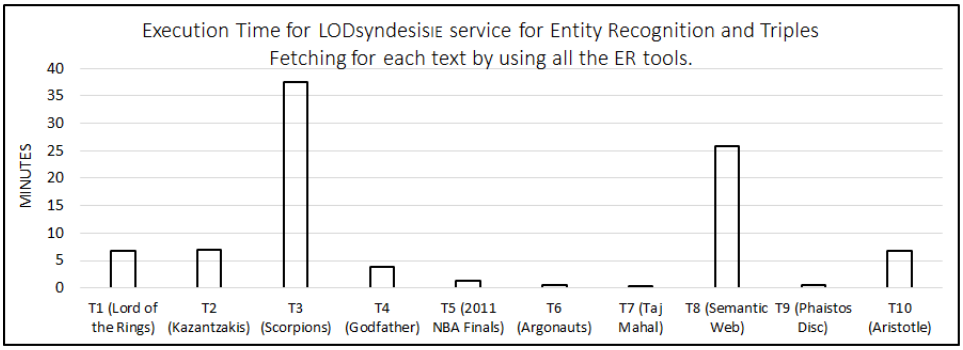

Figure 5. Execution time (in minutes) for LODsyndesis$_{IE}$ Service 3 for ER and Triples Downloading for each text of SimpleWiki collection by using all the ER tools.

4.6. Discussion of Experimental Results

Here, we discuss the results for the hypotheses *h1* and *h2*. As regards Hypothesis H1, for all the evaluation collections, we obtained the highest recall by combining all the ER tools. Concerning the SimpleWiki collection, which contains simple texts, in all cases by combining more ER tools, the values of the recall and the F1 score metrics increased, whereas the value of precision remained high. On the other hand, for the collections containing larger and more complex texts (i.e., AQUAINT and MSNBC), we obtained the highest precision by using a single ER tool and the highest F1 score by either a single or a pair of ER tools. However, by combining the results of the 80 texts of all three evaluation collections, we obtained that by using all the ER tools, we can correctly recognize more entities in the text (i.e., recall increases) and we achieved the highest average F1 Score. On the contrary, in several cases (especially for complex texts), the precision decreases.

Concerning Hypothesis H2, the measurements revealed the gain of linking the recognized entities to multiple datasets for all the tasks. Indicatively, for Content Enrichment, for the entities of SimpleWiki collection, *LODsyndesis* offers on average 13.2 URIs and 1255.5 triples per entity, whereas each entity can be found on average in 8.5 RDF datasets. For the task of Data Veracity, *LODsyndesis* contains more than 14 million verified facts from at least 2 RDF datasets, whereas for each entity of SimpleWiki collection, on average 9.2 facts were verified from *LODsyndesis*. Finally, for the tasks of Data Discovery and Integration, we showed that through LODsyndesis$_{IE}$ it is feasible and fast to discover and select the top-*K* RDF datasets for one or more entities of any given text.

Finally, regarding the efficiency, by using all the ER tools, the execution time was low, i.e., less than 7 s on average for most of the LODsyndesis$_{IE}$ services for the texts of SimpleWiki collection.

5. Concluding Remarks

Since there are no available tools offering *Entity Recognition* (ER) and *Entity Enrichment* through multiple datasets at the same time, we presented the research prototype LODsyndesis$_{IE}$. In particular, we presented an algorithm that combines widely used Entity Recognition tools (i.e., DBpedia Spotlight, Stanford CoreNLP and WAT) for recognizing the entities of a given text. Concerning the algorithm, each ER tool separately identifies the entities of the given text and assigns to them a URI to a knowledge base (e.g., DBpedia). Afterwards, for combining the results of the ER tools and for keeping a single URI for

each identified entity, it uses two rules for deciding which candidate URI will be selected. Regarding the first rule (the Majority Rule), if the majority of tools provide the same URI for an entity, then that URI is selected. Regarding the second rule (the Closest Name Rule), if the tools disagree about the URI of a given entity, the algorithm selects the URI with the minimum Levenhstein distance with the desired entity.

Afterwards, we showed how we annotate the identified entities of the text with links to the *LODsyndesis* knowledge base for linking each entity to hundreds of RDF datasets, i.e., *LODsyndesis* contains advanced services for 400 million entities from 400 RDF datasets. We described how $\texttt{LODsyndesis}_{IE}$ exploits *LODsyndesis*for offering services for the recognized entities for several tasks, including services for Annotation, Hyperlink Creation, Content Enrichment, Data Discovery and Selection, Data Veracity and Integration. Moreover, we provided details about how to use $\texttt{LODsyndesis}_{IE}$ through either its web application or its REST API.

Concerning evaluation, we used three evaluation collections. In particular, we have built one simple collection, called SimpleWiki, for evaluating the effectiveness of combining different ER tools and of linking the entities to hundreds of RDF datasets, whereas we also used two existing collections, i.e., AQUAINT and MSNBC, which contain larger and more complex texts, for evaluating the performance of ER tools. Regarding the effectiveness of Entity recognition process, by taking into account all the texts of the evaluation collections, we obtained the maximum average recall and F1 score by combining the three ER tools. On the contrary, in many cases (especially for more complex texts), the precision decreases by using two or three ER tools.

Regarding measurements for Data enrichment, the results for the SimpleWiki collection revealed the gain of using multiple datasets instead of a single or a few ones. Indicatively, we managed to have access on average to 208.7% more triples for each recognized entity by using *LODsyndesis* versus a single dataset (i.e., DBpedia) and 411.1% more verified facts by using *LODsyndesis* versus DBpedia and Wikidata. Concerning efficiency, most of the services are quite fast; indicatively, $\texttt{LODsyndesis}_{IE}$ needs on average less than 6 seconds for recognizing the entities of a text and for retrieving all the URIS and datasets of each recognized entity.

As a future work, we plan to extend $\texttt{LODsyndesis}_{IE}$ for covering more tasks of the Information Extraction process. One direction is to enable the extraction and annotation of the relations of the given text, for linking them with the corresponding properties to *LODsyndesis*. Another direction is to automate the fact checking process, i.e., to annotate the text with the facts that can be verified through *LODsyndesis*. Moreover, we plan to propose variations of the presented algorithm for improving the precision of the Entity Recognition process by combining many ER tools. Finally, one interesting direction would be to support and to combine even more ER tools.

Author Contributions: Conceptualization, M.M. and Y.T.; data curation, M.M.; investigation, M.M. and Y.T.; methodology, M.M. and Y.T.; project administration, Y.T.; software, M.M.; supervision, Y.T.; writing—original draft, M.M. and Y.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. More Experiments for Entity Recognition Process

Here, we provide the box plots for each evaluation collection for the metrics of precision, recall and F1 score.

Appendix A.1. The Box Plots for SimpleWiki Evaluation Collection

Figures A1–A3 shows the box plots for SimpleWiki collection, for the metrics of precision, recall and F1 score, respectively. Through these figures, we can see the gain of

combining multiple ER tools for the texts of SimpleWiki collection, especially for increasing the recall and the F1 score.

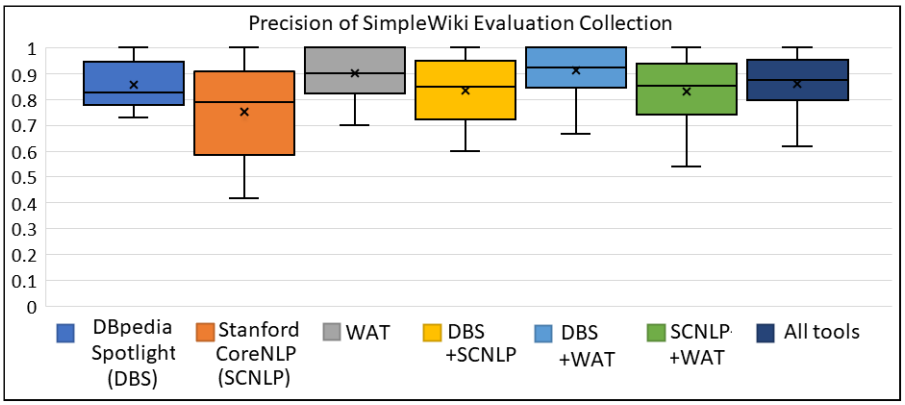

Figure A1. The box plot of Precision for SimpleWiki Evaluation Collection.

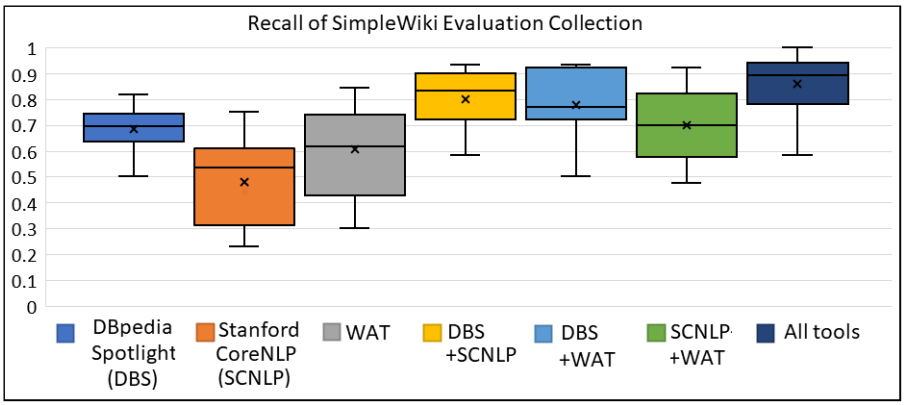

Figure A2. The box plot of Recall for SimpleWiki Evaluation Collection.

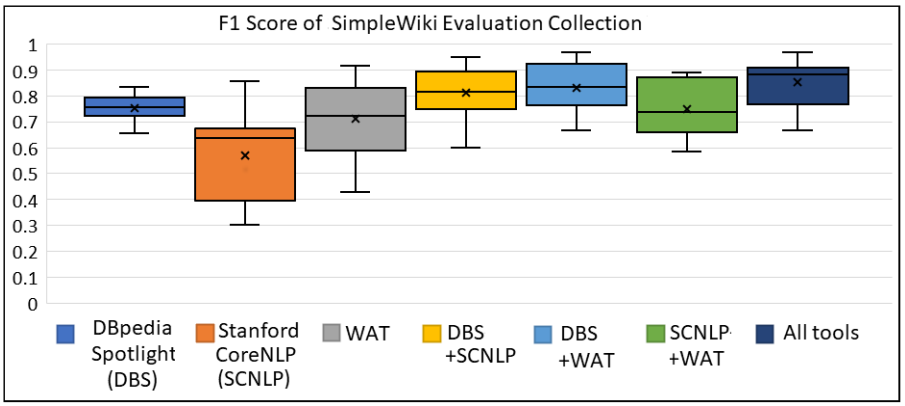

Figure A3. The box plot of F1 Score for SimpleWiki Evaluation Collection.

Appendix A.2. The Box Plots for MSNBC Evaluation Collection

Figures A4–A6 show the box plots for the MSNBC collection for the metrics of precision, recall and F1 score, respectively. We can observe that the WAT tool was quite efficient for the texts of that collection. On the contrary, we can see the increase for the average value of the recall metric by combining the three ER tools.

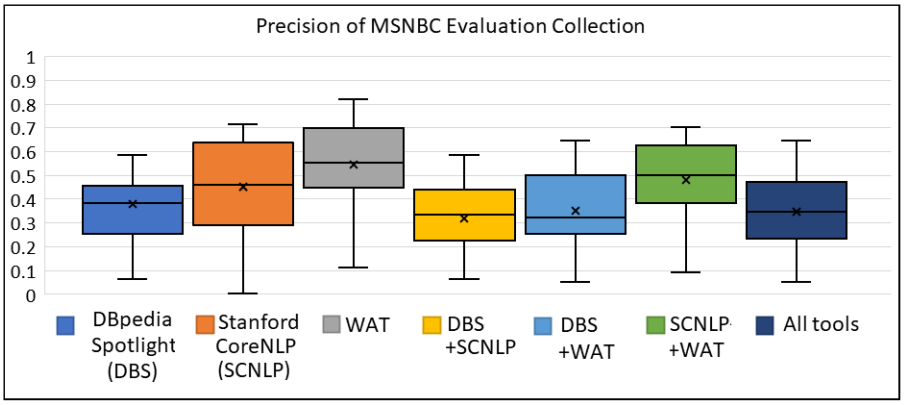

Figure A4. The box plot of Precision for MSNBC Evaluation Collection.

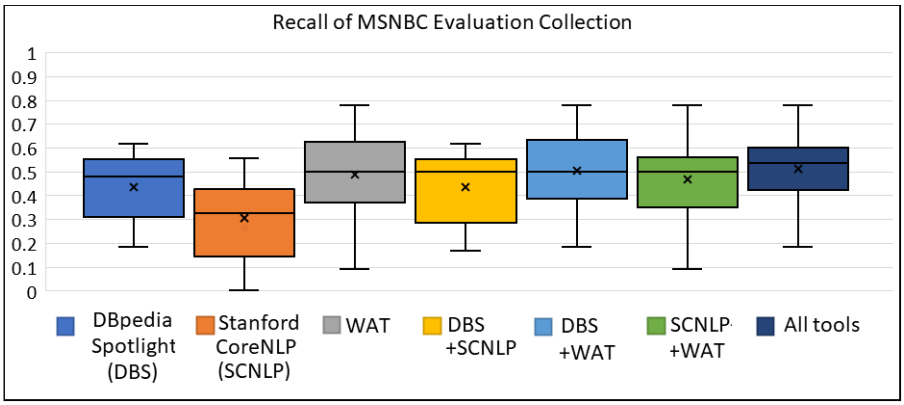

Figure A5. The box plot of Recall for MSNBC Evaluation Collection.

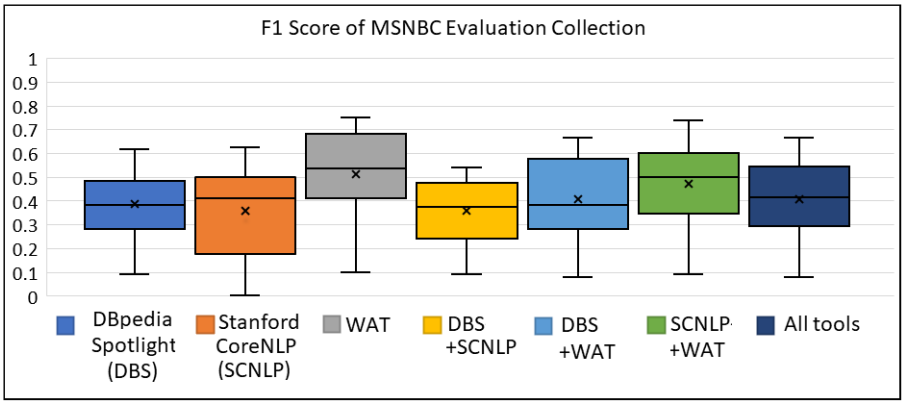

Figure A6. The box plot of F1 Score for MSNBC Evaluation Collection.

Appendix A.3. The Box Plots for AQUAINT Evaluation Collection

Figures A7–A9 show the box plots for the AQUAINT collection for the metrics of precision, recall and F1 score, respectively. We can observe that the tools WAT and SCNLP offered higher precision, whereas DBS achieved higher recall than these two tools. On the contrary, we can see the gain for the metrics of recall (mainly) and F1 score by combining either two or three ER tools.

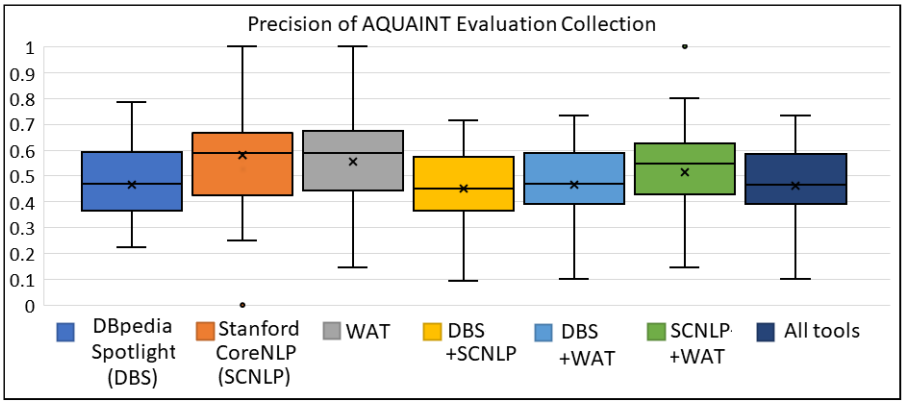

Figure A7. The box plot of Precision for AQUAINT Evaluation Collection.

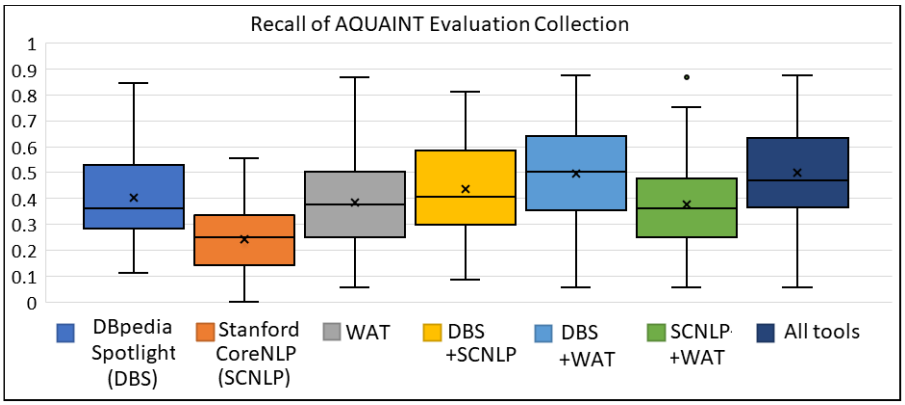

Figure A8. The box plot of Recall for AQUAINT Evaluation Collection.

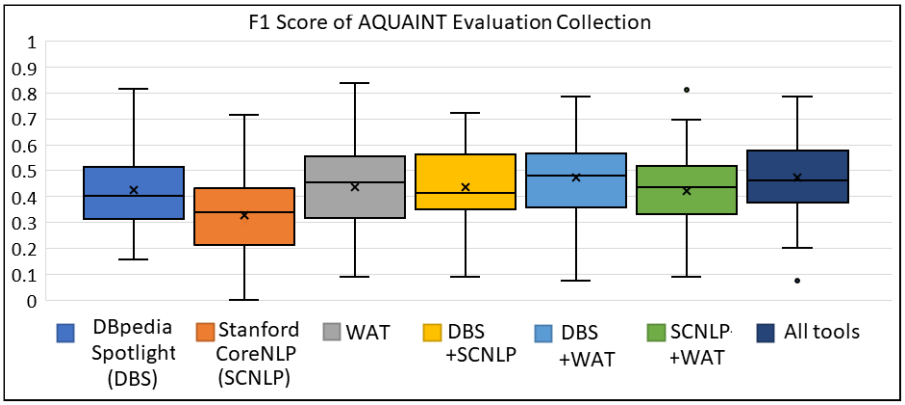

Figure A9. The box plot of F1 Score for AQUAINT Evaluation Collection.

References

1. Grishman, R. Information extraction: Techniques and challenges. In *International Summer School on Information Extraction*; Springer: Cham, Switzerland, 1997; pp. 10–27.
2. Sarawagi, S. *Information Extraction*; Now Publishers Inc.: Delft, The Netherland, 2008.
3. Ermilov, I.; Lehmann, J.; Martin, M.; Auer, S. LODStats: The data web census dataset. In *International Semantic Web Conference*; Springer: Cham, Switzerland, 2016; pp. 38–46.
4. Mountantonakis, M. *Services for Connecting and Integrating Big Numbers of Linked Datasets*; IOS Press: Amsterdam, The Netherland, 2021.
5. Röder, M.; Usbeck, R.; Ngonga Ngomo, A.C. Gerbil–benchmarking named entity recognition and linking consistently. *Semant. Web* **2018**, *9*, 605–625. [CrossRef]

6. Mendes, P.N.; Jakob, M.; García-Silva, A.; Bizer, C. DBpedia spotlight: shedding light on the web of documents. In *SEMANTiCS*; ACM: New York, NY, USA, 2011; pp. 1–8.
7. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*; Springer: Cham, Switzerland, 2007; pp. 722–735.
8. Piccinno, F.; Ferragina, P. From TagME to WAT: A new entity annotator. In Proceedings of the Workshop on Entity Recognition & Disambiguation, Gold Coast, QLD, Australia, 11 July 2014; pp. 55–62.
9. Mountantonakis, M.; Tzitzikas, Y. LODsyndesis: global scale knowledge services. *Heritage* **2018**, *1*, 335–348. [CrossRef]
10. Beek, W.; Raad, J.; Wielemaker, J.; van Harmelen, F. sameAs.cc: The Closure of 500M owl: Same As Statements. In *European Semantic Web Conference*; Springer: Cham, Switzerland, 2018; pp. 65–80.
11. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.R.; Bethard, S.; McClosky, D. The Stanford CoreNLP natural language processing toolkit. In Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 23–24 June 2014; pp. 55–60.
12. Mountantonakis, M.; Tzitzikas, Y. LODsyndesisIE: Entity Extraction from Text and Enrichment Using Hundreds of Linked Datasets. In *European Semantic Web Conference*; Springer: Cham, Switzerland, 2020; pp. 168–174.
13. Bizer, C.; Heath, T.; Berners-Lee, T. Linked data: The story so far. In *Semantic Services, Interoperability and Web Applications: Emerging Concepts*; IGI Global: Pennsylvania, PA, USA, 2011; pp. 205–227.
14. Bechhofer, S.; Van Harmelen, F.; Hendler, J.; Horrocks, I.; McGuinness, D.L.; Patel-Schneider, P.F.; Stein, L.A. OWL web ontology language reference. *W3C Recomm.* **2004**, *10*, 1–53.
15. Rebele, T.; Suchanek, F.; Hoffart, J.; Biega, J.; Kuzey, E.; Weikum, G. YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *International Semantic Web Conference*; Springer: Cham, Switzerland, 2016; pp. 177–185.
16. Moro, A.; Cecconi, F.; Navigli, R. Multilingual Word Sense Disambiguation and Entity Linking for Everybody. In *International Semantic Web Conference (Posters & Demos)*; CEUR-WS.org: Aachen, Germany, 2014; pp. 25–28.
17. Hoffart, J.; Yosef, M.A.; Bordino, I.; Fürstenau, H.; Pinkal, M.; Spaniol, M.; Taneva, B.; Thater, S.; Weikum, G. Robust disambiguation of named entities in text. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, UK, 27–31 July 2011; pp. 782–792.
18. van Hulst, J.M.; Hasibi, F.; Dercksen, K.; Balog, K.; de Vries, A.P. Rel: An entity linker standing on the shoulders of giants. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Xi'an, China, 25–30 July 2020; pp. 2197–2200.
19. Kolitsas, N.; Ganea, O.E.; Hofmann, T. End-to-End Neural Entity Linking. In Proceedings of the 22nd Conference on Computational Natural Language Learning, Brussels, Belgium, 31 October–1 November 2018; pp. 519–529.
20. Martinez-Rodriguez, J.L.; Hogan, A.; Lopez-Arevalo, I. Information extraction meets the semantic web: a survey. *Semant. Web* **2020**, *11*, 255–335. [CrossRef]
21. Al-Moslmi, T.; Ocaña, M.G.; Opdahl, A.L.; Veres, C. Named Entity Extraction for Knowledge Graphs: A Literature Overview. *IEEE Access* **2020**, *8*, 32862–32881. [CrossRef]
22. Singh, K.; Lytra, I.; Radhakrishna, A.S.; Shekarpour, S.; Vidal, M.E.; Lehmann, J. No one is perfect: Analysing the performance of question answering components over the dbpedia knowledge graph. *J. Web Semant.* **2020**, *65*, 100594. [CrossRef]
23. Diefenbach, D.; Singh, K.; Maret, P. WDAqua-core1: A Question Answering service for RDF Knowledge Bases. In Proceedings of the Companion Web Conference 2018, Lyon, France, 23–27 April 2018; pp. 1087–1091.
24. Bast, H.; Haussmann, E. More accurate question answering on freebase. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, VIC, Australia, 18–23 October 2015; pp. 1431–1440.
25. Shekarpour, S.; Marx, E.; Ngomo, A.C.N.; Auer, S. Sina: Semantic interpretation of user queries for question answering on interlinked data. *J. Web Semant.* **2015**, *30*, 39–51. [CrossRef]
26. Dimitrakis, E.; Sgontzos, K.; Mountantonakis, M.; Tzitzikas, Y. Enabling Efficient Question Answering over Hundreds of Linked Datasets. In *International Workshop on Information Search, Integration, and Personalization*; Springer: Cham, Switzerland, 2019; pp. 3–17.
27. Dimitrakis, E.; Sgontzos, K.; Tzitzikas, Y. A survey on question answering systems over linked data and documents. *J. Intell. Inf. Syst.* **2019**, *55*, pp. 233–259. [CrossRef]
28. Maliaroudakis, E.; Boland, K.; Dietze, S.; Todorov, K.; Tzitzikas, Y.; Fafalios, P. ClaimLinker: Linking Text to a Knowledge Graph of Fact-checked Claims. In Proceedings of the Companion Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 669–672.
29. Chabchoub, M.; Gagnon, M.; Zouaq, A. Collective disambiguation and semantic annotation for entity linking and typing. In *Semantic Web Evaluation Challenge*; Springer: Cham, Switzerland, 2016; pp. 33–47.
30. Beno, M.; Filtz, E.; Kirrane, S.; Polleres, A. Doc2RDFa: Semantic Annotation for Web Documents. In Proceedings of the Posters and Demo Track of the 15th International Conference on Semantic Systems, Karlsruhe, Germany, 9–12 September 2019; CEUR-WS.org: Aachen, Germany, 2019.
31. Xiong, C.; Callan, J.; Liu, T.Y. Word-entity duet representations for document ranking. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 763–772.
32. Geiß, J.; Spitz, A.; Gertz, M. Neckar: A named entity classifier for wikidata. In *International Conference of the German Society for Computational Linguistics and Language Technology*; Springer: Cham, Switzerland, 2017; pp. 115–129.

33. Vrandečić, D.; Krötzsch, M. Wikidata: a free collaborative knowledge base. *Commun. ACM* **2014**, *57*, 78–85. [CrossRef]
34. Sakor, A.; Singh, K.; Patel, A.; Vidal, M.E. Falcon 2.0: An entity and relation linking tool over Wikidata. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Xi'an, China, 19–23 October 2020; pp. 3141–3148.
35. Noullet, K.; Mix, R.; Färber, M. KORE 50DYWC: An Evaluation Data Set for Entity Linking Based on DBpedia, YAGO, Wikidata, and Crunchbase. In Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, 11–16 May 2020; pp. 2389–2395.
36. Valdestilhas, A.; Soru, T.; Nentwig, M.; Marx, E.; Saleem, M.; Ngomo, A.C.N. Where is my URI? In *European Semantic Web Conference*; Springer: Cham, Switzerland, 2018; pp. 671–681.
37. Mountantonakis, M.; Tzitzikas, Y. Content-Based Union and Complement Metrics for Dataset Search over RDF Knowledge Graphs. *J. Data Inf. Qual. (JDIQ)* **2020**, *12*, 1–31. [CrossRef]
38. Beek, W.; Rietveld, L.; Bazoobandi, H.R.; Wielemaker, J.; Schlobach, S. LOD laundromat: A uniform way of publishing other people's dirty data. In *International Semantic Web Conference*; Springer: Cham, Switzerland, 2014; pp. 213–228.
39. Fernández, J.D.; Beek, W.; Martínez-Prieto, M.A.; Arias, M. LOD-a-lot. In *International Semantic Web Conference*; Springer: Cham, Switzerland, 2017, pp. 75–83.
40. Acosta, M.; Hartig, O.; Sequeda, J. Federated RDF query processing. In *Encyclopedia of Big Data Technologies*; Sakr S., Zomaya A., Eds.; Springer: Cham, Switzerland, 2018; pp. 1–8.
41. Guha, R.V.; Brickley, D.; Macbeth, S. Schema. org: evolution of structured data on the web. *Commun. ACM* **2016**, *59*, 44–51. [CrossRef]
42. Mountantonakis, M.; Tzitzikas, Y. Large Scale Semantic Integration of Linked Data: A survey. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 103. [CrossRef]
43. Cucerzan, S. Large-scale named entity disambiguation based on Wikipedia data. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic, 28–30 June 2007; pp. 708–716.
44. Milne, D.; Witten, I.H. Learning to link with wikipedia. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, CA, USA, 26–30 October 2008; pp. 509–518.