An Approach for Populating and Enriching Ontology-based Repositories

Alda Canito, Paulo Maio and Nuno Silva
School of Engineering, Polytechnic of Porto
Rua Dr. António Bernardino de Almeida, 431
4200-072 Porto
{alrfc, pam, nps}@isep.ipp.pt

Abstract – **Publically available text-based documents (e.g. news, meeting transcripts) are a very important source of knowledge, especially for organizations. These documents mention domain entities such as persons, places, professional positions, decisions and actions. Querying these documents (instead of browsing, searching and finding) is a very relevant task for any person in general, and particularly for professionals dealing with intensive knowledge tasks. Querying text-based documents' data, however, is not supported by common technology. For that, such documents' content has to be explicitly and formally captured as facts into a knowledge base. Making use of automatic NLP processes for capturing such facts is a common approach, but their relatively low precision and recall give rise to data quality problems. Furthermore, facts existing in the documents are often insufficient to answer complex queries, thus the need to enrich the captured facts with facts from third-party repositories (e.g. public LOD). This paper describes the adopted process to clean, populate and enrich a knowledge base repository that is further exploited to answer complex queries. This process is triggered by a previous NLP parsing process and conducted by the (rich) ontology describing such repository.**

Keywords – Ontology Population, Ontology Data Enrichment, Ontology-based Data Cleaning

I. INTRODUCTION

Publically available text-based documents (e.g. news, meeting transcripts) are a very important source of knowledge, especially for organizations. Querying the content of these documents is not technologically supported, which forces the user to search, browse and integrate information by him/herself. This is a time-consuming, tedious, error-prone, unrepeatable and unconfident process.

It is our aim to create a system that is able to address semantically rich and complex queries over the content of unstructured or semi-structured documents [1]. While the repository, the query building and execution applications are now available and functional [1], to maintain and populate the repository with facts from unstructured and semi-structured documents is still an open issue. In particular, on those documents it is necessary to identify ontological instances (e.g. places, persons, professional positions, decisions, and actions) and their relationships.

Making use of automatic NLP processes is a common approach for capturing and explicitly and formally "semantizing" the documents' content. However, the relatively low precision and recall of the automatic NLP processes [2], [3] give rise to data quality problems, including duplicates, incoherencies, inconsistencies and incompleteness.

Currently, to avoid (or at least to minimize) those data quality problems, facts output by the NLP process are conveniently analyzed and processed manually by users before being integrated into the repository.

During this user-based process, it has been perceived as a major issue: the NLP' acquired data is often insufficient for the purpose of applying a reasoner [4] (or a classifier) to infer new facts from the already known data (A-Box) together with the terminological component (T-Box) of the repository.

In order to (i) automatize the aforementioned user-based process and (ii) to address the incompleteness of the data, this paper proposes a novel iterative and incremental process that combines simultaneously tasks as (a) data cleaning, (b) ontology population and (c) enrichment for inference purposes. The proposed process is therefore triggered by the NLP process generated data and driven by the semantically rich OWL [5] ontology describing the repository where such data is intended to be integrated. In particular, the enrichment task seeks on third-parties repositories the missing data that will promote inference (namely classification of instances).

The rest of this paper is organized as follows. Next section presents the technological context underlying this research. Section 3 presents a formal description of the proposed process called Ontology-driven Data Cleansing and Enrichment (ODCE). Our proposal is compared to other works in section 4. Finally, section 5 summarizes the contributions, draws some conclusions and points out next research steps.

II. CONTEXT

The work presented in [1] describes the system architecture of the World Search project [6] and the Complex Query Building (CQB) process [1] (Figure 1). The architecture has six main modules with the following responsibilities:

- GUI is responsible for the interaction with the user: (i) it enables the user to formulate queries and (ii) presents the respective responses (i.e. the retrieved results);
- Query Dispatcher is responsible for forwarding the queries to the proper answering module;

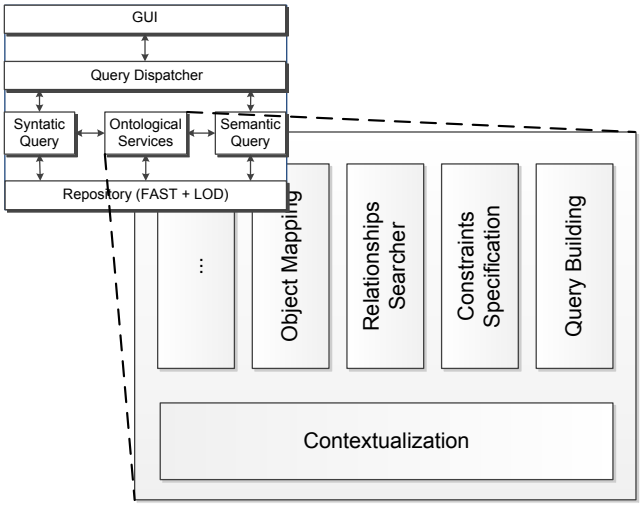

Figure 1. The World Search system architecture.

- Ontological Services module is responsible for managing the semantic information of the system (e.g. maintain the ontologies underlying the system) and providing semantic services (e.g. synonyms correspondences between concepts) to other system' modules. Further, it has been filled in with core competencies of the CQB and its contextualization module [7];
- Syntactic Query module is responsible for retrieving resources based on a text-based search only. It might make use of the ontological services to expand the query based on the synonyms of the words specified in it;
- Semantic Query module is responsible for retrieving resources based on the semantic entities specified by the user. The ontological services are exploited in order to perceive the relations between those entities and other ontological entities. The CQB approach exploits the semantics and structure of the ontology(ies) in order to support the user creating complex queries;
- Repository is where all the data/information supporting both the syntactic and the semantic queries is maintained. Currently, this module is mainly composed by (i) a syntactic search engine (e.g. FAST, Lucene) for indexing purposes and (ii) a data source repository meeting the Linked Open Data principles [8].

In the following, it is described the new process lying in the ontological services module for cleaning, populating and enriching the repository with data that promotes and allows inference of new data, driven by a semantically rich OWL ontology.

III. THE ODCE PROCESS

This section describes the proposed Ontology-driven Data Cleansing and Enrichment (ODCE) process independently from any concrete implementation or application domain.

The ODCE process consists of eight steps (or tasks) as depicted in Figure 2.

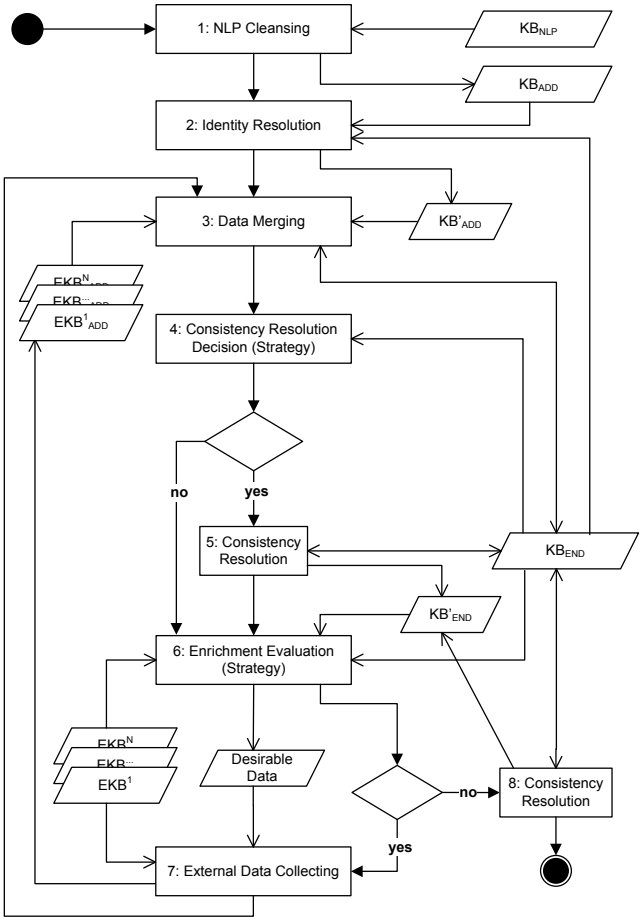

Figure 2. The ODCE process.

Consider KB_{NLP} the output of a NLP parsing process over a set of documents, which consists on a set of facts represented according to a lightweight ontology (O_{NLP}). Yet, consider that these facts need to be integrated and enriched in a proper, consistent and automatically way into another knowledge base (KB_{END}) whose content is described according to an ontology (O_{END}). This ontology (O_{END}) captures the same domain knowledge of O_{NLP} but, it is semantically richer (for inference purposes) and, contrary to O_{NLP}, it also takes into consideration ontological orthogonal dimensions such as time and space.

The first step aims to ensure KB_{NLP} is consistent and ready for integration. For that, it applies a set of data cleansing operations to detect and to correct inaccurate information, namely to avoid duplicated facts and/or entities. The result of this step is a minimal, clean and consistent set of facts (KB_{ADD}) to be integrated into the KB_{END} knowledge base. The aforementioned cleansing task is formalized as follows.

Definition 1 (Cleansing Task) – The cleansing task of the information contained by a single knowledge base is seen as a function $cleansing(KB) \rightarrow KB'$ such that KB

refers to the knowledge base on which the cleansing operation occurs and KB' refers to a modified version of KB such that its content is clean and consistent.

The second step consists in identifying univocally the entities mentioned in KB_{ADD} regarding the knowledge base on which information will be integrated (KB_{END}). Thus, it is necessary to verify, for each entity $e \in KB_{ADD}$, if exists an entity $e' \in KB_{END}$ such that e and e' are considered by a given identity function as referring to the same real/domain entity. In such cases, all references to e into KB_{ADD} are replaced by e' giving raise to KB'_{ADD}. An identity function is formalized as follows.

Definition 2 (Identity Function) – An identity function is defined as $f_{ID}(KB_S, e, KB_T) \rightarrow E$ such that:

- KB_S is the knowledge base defining and containing facts about the entity e;
- $e \in KB_S$ is the entity on which the function acts in order to find another known identity;
- KB_T is the knowledge base on which the function searches for an already existing entity that can be considered as referring to the same entity of e;
- E is the resulting entity such that it is either (i) entity e in case there is no entity into KB_T considered as being the same or (ii) an entity $e' \in KB_T$ otherwise.

Considering the above definition, the entities' identity resolution task is formalized as follows.

Definition 3 (Identity Resolution Task) – The entities' identity resolution existing in a knowledge base according to the content of another knowledge base is seen as a function $idresolution(KB_S, KB_T, f_{ID}) \rightarrow KB'_S$ such that:

- KB_S is the (source) knowledge base containing the entities on which it is necessary to verify and resolve their identity (e.g. KB_{ADD});
- KB_T is the (target) knowledge base where to search for an already existing identity (e.g. KB_{END});
- f_{ID} is the identity function used to search the target knowledge base in order to find the existing entity's identity;
- KB'_S is the resulting knowledge base containing the same facts of KB_S but whose entities' identity is in accordance with KB_T (e.g. KB'_{ADD}).

The third step consists on taking several knowledge bases (namely KB'_{ADD}) containing previously collected and prepared information with the unique purpose of being integrated (or merged) into the target knowledge base (KB_{END}). Therefore, it entails that a transparent data transformation process occurs between each (possible) pair of source and target knowledge bases. In this sense, a data transformation process is formalized as follows.

Definition 4 (Data Transformation) – The data transformation process between two knowledge bases (source and target) is seen as a function $T(KB_S, KB_T, A) \rightarrow KB'_T$ such that:

- KB_S is the source knowledge base (e.g. KB'_{ADD}) containing the information to be transformed and integrated on KB_T;
- KB_T is the target knowledge (e.g. KB_{END}) base on which the transformed information will be added (or updated);
- A is a (declarative) alignment between the ontologies describing the source and target knowledge bases on which the data transformation process relies on;
- KB'_T corresponds to the target knowledge base updated with the integrated facts.

Considering the above definition, the data merging task is formalized as follows.

Definition 5 (Data Merging Task). The data merging (or integration) task between a set of source knowledge based and a target knowledge base is seen as a function $\mathfrak{T}(S, KB_T, \mathcal{A}, mapS, dt) \rightarrow KB'_T$ such that:

- S is a set of knowledge bases containing the information to be integrated into KB_T;
- KB_T is the target knowledge base on which the information will be integrated (i.e. added or updated);
- $\mathcal{A}$ is a set of (declarative) alignments between the ontologies describing the source and target knowledge bases;
- $map\mathcal{A}: S \rightarrow \mathcal{A}$ is a function mapping each source knowledge base in S to an alignment $A \in \mathcal{A}$ on which the data transformation will rely on;
- dt corresponds to a data transformation process established according to **Definition 4**.
- KB'_T corresponds to the target knowledge base updated with the integrated facts.

The result of the data merging task may leave the KB_{END} knowledge base temporarily inconsistent. The next steps are responsible for the resolution of those inconsistencies.

The fourth step checks if the KB_{END} knowledge base has inconsistencies resulting from the execution of the previous steps. Since the knowledge base is described by an OWL ontology (and some SWRL rules), inconsistencies can be analyzed and identified by means of a reasoner such as Pellet [4].

Based on (i) the (in)existence of inconsistencies and (ii) on the kind of inconsistencies found, it decides either:

- To resolve the inconsistencies found and, therefore, the ODCE process flows to step five before doing the enrichment;
- To proceed immediately to the enrichment task (step 6) since the inconsistencies found are not considered as causing undesirable effects (e.g. malfunctioning, incompleteness) on the enrichment task.

It is worth to bear in mind that this decision should also take into consideration other issues such as the performance of the overall process, the requirements of the enrichment process, the interdependencies between the ODCE process and the running application (e.g. answering complex question) that rely on the knowledge base. Yet, it is important noticing that, independently of the decision made at this point, the ODCE process ensures that, at the end, the

knowledge base is consistent. This is achieved because the last step of the process corresponds to a mandatory consistency resolution task (see below).

The purpose of the sixth step is to evaluate the need to enrich the KB_{END} knowledge base with new (missing) facts that foster the further (re)classification of the instances (A-Box). These new facts can be represented (or captured) by the ontology describing KB_{END} and should be available on external knowledge bases (e.g. DBPedia). The driving vector of this task is the terminological component (T-Box) of KB_{END} and the underlying semantics of the OWL constructs (e.g. equivalent class, intersection, union, and complement). However, different ontological constraints require different approaches. So, it is envisioned the adoption of different enrichment strategies capturing and considering the following dimensions: (i) the available and required data, (ii) the characteristics of the ontological constraints and (iii) the characteristics of the data sources.

At the end of this step, it has been identified the desirable information that is lacking in KB_{END}. It should be noted that the desirable information may resolve some inconsistencies generated by the third step but may also raise other (new) inconsistencies. This task is formalized as follows.

Definition 6 (Enrichment Evaluation) – The enrichment evaluation task is seen as a function $evalEnrich(KB, ES) \rightarrow \{D, enrich\}$ such that:

- KB is the knowledge base to be enriched;
- ES is a set of external knowledge bases that can be used to extract information with the purpose to enrich KB;
- D is an specification of which data/information is desirable to extract from the external knowledge bases;
- $enrich \in \{true, false\}$ instructs the ODCE process either:
  - To proceed to the External Data Collecting step if it has been verified the need to enrich/collect information from the external knowledge bases ($true$);
  - To proceed to the last step (Consistency Resolution) of the ODCE process ($false$);

The seventh step aims to extract/collect from a set of external knowledge bases the information specified in D as desirable. The result is a set of knowledge bases (one for each external knowledge base used) containing the information to be further integrated in the KB_{END} knowledge base. This task is formalized as follows.

Definition 7 (External Data Collecting) – The enrichment process is seen as a function $collecting(D, ES, F, mapES, \mathcal{A}) \rightarrow EKB$ such that:

- D is the specification of the information to collect;
- ES is the set of external knowledge bases from where to collect each $d \in D$;
- F is a set of identity functions as defined in **Definition 2**;
- $mapF: ES \rightarrow F$ is a function that maps each $ekb \in ES$ to an identity function that is used to identify the entities in the ekb that have a similar identity to the ones mentioned in D;
- $\mathcal{A}$ is a set of (declarative) alignments between the desirable data knowledge base and the ontologies describing the external data sources;
- $map\mathcal{A}: ES \rightarrow \mathcal{A}$ is a function mapping each external data source knowledge base in ES to the alignment in $\mathcal{A}$ on which the data collecting process will rely on;
- EKB is a set of knowledge bases containing the collected data, where each $ekb' \in EKB$ is related to an external knowledge base $ekb \in ES$ such that ekb' is a sub-set of ekb ($ekb' \subseteq ekb$).

At the end of this task, the process proceeds again to the Data Merging step in order to merge the collected data into the KB_{END} knowledge base. After the collected data is merged new enrichment desires may appear. As that, a new iteration of the process starts. The process has as many iterations as necessary.

At the end of the process and in order to ensure that KB_{END} knowledge base is consistent, the last step is a mandatory Consistency Resolution task. This task consists in identifying and resolving the inconsistencies caused by the information added (or modified) on the KB_{END} knowledge base by the Data Merging task. This task is formalized as follows.

Definition 8 (Consistency Resolution) – The task to ensure consistency of a knowledge base is seen as a function $consistency(KB) \rightarrow \{KB', KB_{In}\}$ such that:

- KB refers to the knowledge base (possibly) containing inconsistencies;
- KB' corresponds to a knowledge base which is a modified version of KB where the inconsistencies have been resolved;
- KB_{In} is a complementary knowledge base containing the information that has been removed from KB in order to ensure its consistency.

Steps regarding the enrichment tasks (six and seven respectively) are the most relevant and innovative ones in our proposal. As such, comparisons with other work provided in the next section mainly consider these steps.

IV. Related Work

Ontology enrichment is commonly used to describe the process of parsing documents to extract schemas and facts. Some approaches, like the system described in [2], aim to extract and expand an ontology using several sources of structured and unstructured data (e.g. web documents, databases) using NLP. This system addresses the problem of heterogeneity of data and attempts to solve it by providing a framework which extracts ontologies from a set of data. Much like our proposal, the enrichment process is further enhanced by scanning other data sources. However, and contrary to our work, it is focused on expanding the range of concepts that the ontology covers.

The system described in [9] extracts instances from various sources through a domain-independent ontology-driven approach. Like our system, the T-Box drives the

population process, determining which data to extract. But, unlike our system, it makes no use of the existing instances and, therefore, it does not attempt to fill holes in the existing data.

In [10] the authors present a system whose goal is to mix traditional information retrieval queries and ontology-based queries. For that, documents are semantically annotated based on a very lightweight ontology through a very NLP-oriented process. In this respect, this system corresponds to the NLP-based parsing component of our approach that triggers the proposed ODCE process. Moreover, while our goal is to enrich the A-Box in order to reason and infer new data enabling complex query answering, the aforementioned system intends to gather data as much and broad as possible to provide extended information.

The work presented in [11] evaluates the type of changes that can occur at the terminological level of an ontology. Depending on the nature of such change the system chooses an appropriate strategy predicting other changes that may be necessary to maintain its consistency. In our proposal, a similar strategy-based approach is used to drive the enrichment process instead of the evolution of the ontology.

V. Conclusions and Future Work

The primary emphasis of the research presented in this paper is focused on proposing a process that minimizes the users' efforts on integrating into a repository described by a semantically rich ontology a set of facts extracted from unstructured documents through an NLP process in a proper and coherent way. The envisaged process aims to address not only the traditional problems found in these kind of scenarios such as existence of duplicated entities (or facts) and inconsistencies, but also add to the repository additional (missing) data considered as desirable for inference purposes, namely instances classification. As that, a set of tasks have been clearly identified, systematized, formalized and combined together into a straightforward iterative and incremental process.

On one hand, the ODCE process comprehends tasks that have already been subject to a lot of research, such as data cleansing, identity resolution, consistency checking, from which one can profit in order to provide a concrete implementation. On the other hand, the ODCE process also comprehends tasks that have received a much lesser research effort such as the identification of missing but desirable facts driven by T-Box axioms promoting inference and, therefore, providing a concrete implementation is not trivial and will deserve our future attention. Considering this, it was developed a prototype of the ODCE process in order to perform some preliminary experiments (not reported due to lack of space). Though, preliminary experiments allow perceiving that:

- Users' effort has been substantially reduced (around 65%) on which concerns to the data cleaning, identity resolution and consistency resolution tasks (not considering issues raised by missing data). This results essentially by adopting methods and algorithms (e.g. for the identity resolution task) having an high precision in disfavor of a higher recall;
- Users' effort exclusively caused by missing data has only a marginal reduction (around 15%). This is essentially due to issues related to accessing third-parties data sources, namely the poor quality of the alignments automatically generated by the adopted state-of-the-art ontology matching tools [12].

Considering the preliminary experiments results, authors will address in the near future two main questions: (i) improve substantially the ontology alignments provided by automatic ontology matching tools, which is intended to be done manually and further generate data source accessors based on those alignments and (ii) systematize the approach taken to identify the desirable data.

Acknowledgment

This work is supported by FEDER Funds through the "Programa Operacional Factores de Competitividade - COMPETE" program and by National Funds through FCT "Fundação para a Ciência e Tecnologia" under the projects: World Search (QREN11495) and AEMOS (PTDC/EIA-EIA/104752/2008).

References

[1] R. Brandão, P. Maio, and N. Silva, 'Enhancing LOD Complex Query Building with Context', presented at the IEEE/WIC/ACM International Conference on Web Intelligence, Macau, China, 2012.

[2] F. Song, G. Zacharewicz, and D. Chen, 'An ontology-driven framework towards building enterprise semantic information layer', *Advanced Engineering Informatics*, vol. 27, no. 1, pp. 38–50, 2013.

[3] N. J. Mamede, J. Baptista, I. Trancoso, and M. das G. V. Nunes, *Computational Processing of the Portuguese Language*. Faro, Portugal: Springer, 2003.

[4] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, 'Pellet: A practical owl-dl reasoner', *Web Semantics: science, services and agents on the World Wide Web*, vol. 5, no. 2, pp. 51–53, 2007.

[5] 'OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition)'. [Online]. Available: http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/.

[6] 'World Search', 2010. [Online]. Available: http://www.microsoft.com/portugal/mldc/worldsearch/en/.

[7] R. Brandão, P. Maio, and N. Silva, 'I3OM - An Iterative, Incremental and Interactive Approach for Ontology Navigation based on Ontology Modularization.', in *KEOD*, 2012, pp. 265–270.

[8] T. Berners-Lee, 'Linked Data - Design Issues', Jul-2006. Available: http://www.w3.org/DesignIssues/LinkedData.html.

[9] L. K. McDowell and M. Cafarella, 'Ontology-driven, unsupervised instance population', *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, no. 3, pp. 218–236, Sep. 2008.

[10] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff, 'Semantic annotation, indexing, and retrieval', *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 2, no. 1, pp. 49–79, Dec. 2004.

[11] L. Stojanovic, A. Maedche, B. Motik, and N. Stojanovic, 'User-driven Ontology Evolution Management', in *13th International Conference on Knowledge Engineering and Knowledge Management*, Heidelberg, 2002, vol. 2473, p. 197—212.

[12] OAEI, 'Ontology Alignment Evaluation Initiative', *2011 Campaign*, 2011. [Online]. Available: http://oaei.ontologymatching.org/2011/.