Demo Application for LETO: Learning Engine Through Ontologies

Suilan Estevez-Velarde[1], **Andrés Montoyo**[2], **Yudivián Almeida-Cruz**[1], **Yoan Gutiérrez**[3], **Alejandro Piad-Morffis**[1], and **Rafael Muñoz**[2]

[1]School of Math and Computer Science, University of Havana, Cuba
{sestevez,yudy,apiad}@matcom.uh.cu
[2]Department of Languages and Computing Systems, University of Alicante, Spain
[3]U.I. for Computer Research (IUII), University of Alicante, Spain
{montoyo,ygutierrez,rafael}@dlsi.ua.es

Abstract

The massive amount of multi-formatted information available on the Web necessitates the design of software systems that leverage this information to obtain knowledge that is valid and useful. The main challenge is to discover relevant information and continuously update, enrich and integrate knowledge from various sources of structured and unstructured data. This paper presents the Learning Engine Through Ontologies (LETO) framework, an architecture for the continuous and incremental discovery of knowledge from multiple sources of unstructured and structured data. We justify the main design decision behind LETO's architecture and evaluate the framework's feasibility using the Internet Movie Data Base (IMDB) and Twitter as a practical application.

1 Introduction

In recent years, research in machine learning, knowledge discovery, data mining and natural language processing, among others, have produced many approaches and techniques to deal with the large amount of information available on the Internet to carry out a variety of tasks, such as, for example building search (Brin and Page, 1998) and recommendation systems (Davidson et al., 2010) that could be used to improve business, health-care and political decisions (Ferrucci et al., 2013).

The purpose of our proposal is to present LETO: Learning Engine Through Ontologies, a framework to automatically and gradually extract knowledge from different sources (both structured and unstructured), building internal representations that can be adapted to and integrated in multiple domains. The current state of LETO's implementation is a computational prototype that illustrates the different components of its architecture and demonstrate its feasibility. Inspired by the different processes that occur during human learning, we design the framework's architecture as a learning pipeline that gradually builds more complex knowledge.

In a simplified view, the human learning process can be modeled as a continuous loop that transforms sensorial data into knowledge (see Figure 1) (Gross, 2015). Humans collect information about the environment through senses, where the human brain attempts to detect relations between individual signals to form a more structured representation of reality. By relating as many signals as possible, humans build a much richer semantic representation of the environment, which is unconsciously filtered storing only the most relevant part. In order to achieve this, the brain is able to access to stored experiences about what has been important before, and what is already known. This feedback loop also evaluates previously known facts, and modifies them at the light of new experiences. In time, humans not only learn new facts, but also learn better ways of learning.

The challenge of building computational knowledge discovery systems is an active research problem in the field of artificial intelligence, specifically in emerging areas such as ontology learning (Cimiano et al., 2009) and learning by reading (Barker et al., 2007). Modern systems employ a combination of knowledge-based techniques (i.e., using rules handcrafted by domain experts (Chandrasekaran, 1986)) and statistical approaches (i.e., based on pattern recognition with statistical and probabilistic models (Kevin, 2012)).

Given the large amount of information available online, several knowledge discovery systems fo-

Proceedings of Recent Advances in Natural Language Processing, pages 276–284,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_032

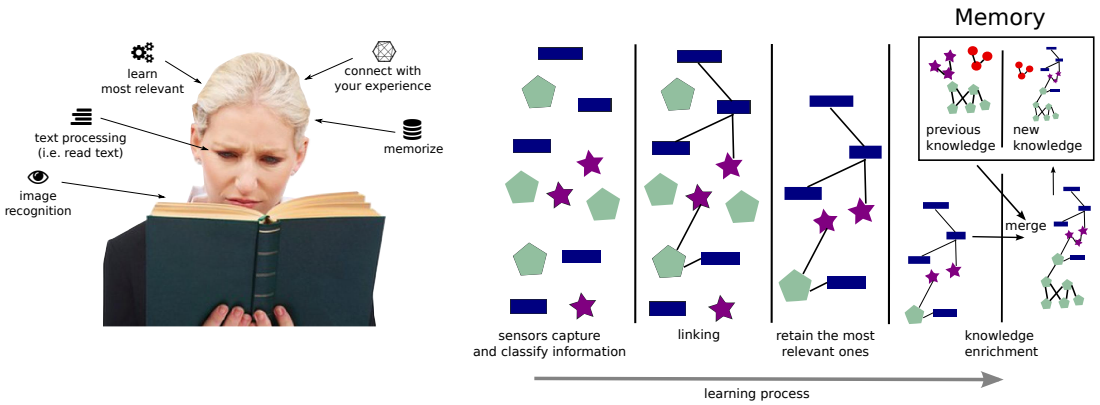

Figure 1: Simplified representation of the human learning process.

cus on extracting knowledge and exploiting the semi-structured format of web resources , e.g, ARTEQUAKT (Alani et al., 2003), SOBA (Buitelaar et al., 2006) and WEB->KB (Craven et al., 2000). In order to extract relevant knowledge from natural language text, NLP techniques have been introduced in systems such as OPTIMA (Kim et al., 2008) and ISODLE (Weber and Buitelaar, 2006). Natural language features can be used to build rule-based systems (e.g., OntoLT (Buitelaar and Sintek, 2004)) or systems based on statistical or probabilistic models trained on NLP corpora, such as LEILA (Suchanek et al., 2006) or Text2Onto (Cimiano and Völker, 2005). Some systems address the issue of inferring more abstract knowledge from the extracted facts, often using unsupervised techniques to discover inherent structures. Relevant examples of this approach are OntoGain (Drymonas et al., 2010), ASIUM (Faure and Poibeau, 2000) and BOEMIE (Castano et al., 2007).

Most of the mentioned systems focus on one iteration of the extraction process. However, more recent approaches, like NELL (Mitchell et al., 2018), attempt to learn continuously from a stream of web data, and increase over time both the amount and the quality of the knowledge discovered.

One of the main characteristics of LETO, in contrast to similar proposals in the literature (such as NELL (Mitchell et al., 2018) or BOEMIE (Petasis et al., 2011)), is the explicit management of separated pieces of knowledge. By isolating the knowledge for different domains, it is possible to apply different techniques and/or parameters as appropriate. Besides, this allows the temporal existence of contradictions or unreliable information that can be crosschecked in the future.

The rest of the paper is organized as follows to facilitate a detailed description of our proposal: Section 2 describes the proposed architecture of a general framework for knowledge discovery. In Section 3 we present an application of LETO to a specific knowledge discovery problem combining Twitter and IMDB. Finally, in Section 4 we present the main conclusions of the research and outline possible future works.

2 Learning Engine Through Ontologies (LETO)

In this section we present LETO, a general architecture for a framework designed to discover relevant knowledge from a variety of data sources, both structured and unstructured.

The LETO framework is divided into 6 modules, which are interrelated. Each module has a specific responsibility defining the inputs and outputs that establish the intercommunication among the rest of the modules within the framework. Figure 2 shows a general overview of the framework.

As shown in Figure 2 the top layer (Data Sources) represents the sources of data that serve as input for the framework. The middle layer contains the Main Modules, which perform the processing of the input data to extract and discover the relevant knowledge latent in this data. Figure 2 also shows the subprocesses that occur inside each module. The main modules always communicate with each other by sharing ontologies. The following sections 2.1, 2.2 and 2.3 explain in detail the inner workings of the main modules. The bottom layer (Backend) contains modules used by the rest of framework:

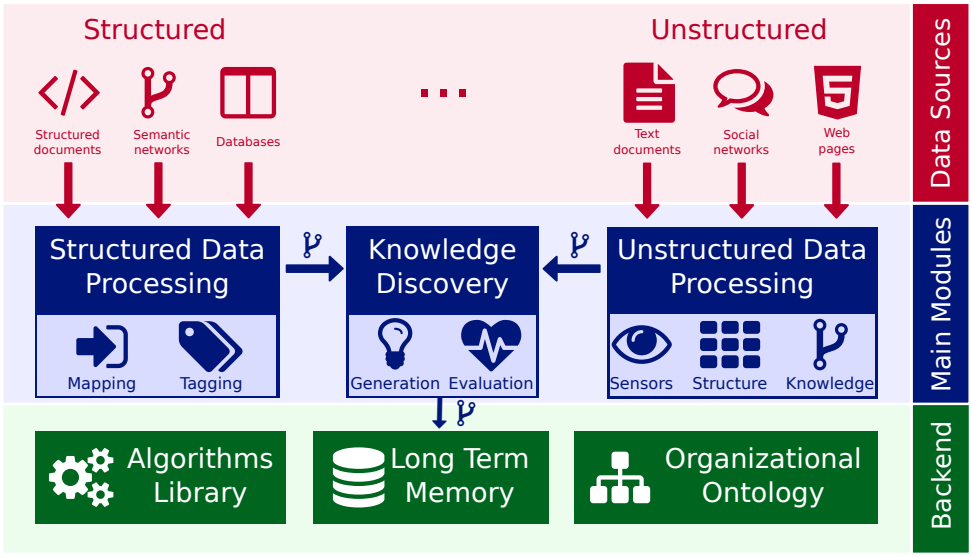

Figure 2: Overview of the architecture of LETO.

Algorithms Library: Contains different algorithms and mathematical models for solving specific problems, along with associated metadata.

Long Term Memory: Contains all the knowledge accumulated by the other modules, in the form of individual ontologies with metadata that describes their content.

Organizational Ontology: An internal representation of the framework's components in an ontological format which enables the automatic construction of the user interface.

2.1 Structured Data Processing

This module is responsible for processing structured data. Sources for structured data are available online in different formats. Among the different types of structures for representing information, such as relational databases, concept maps, knowledge graphs, and others, LETO proposes the use of ontologies for their semantic richness. Ontologies were chosen because they are more expressive than other DTO (Data Transfer Object) formats.

The general pipeline that this module performs can be thought of as a classic Extract, Transform and Load process (ETL) (Vassiliadis, 2009; Hermida et al., 2012). Afterwards, the normalized and tagged block of knowledge (stored as an ontology) is handled to the knowledge processing module, for further refinement and storage. Figure 3 shows an schema of this module. This module performs two main tasks:

Mapping: Since there are many different structured formats, the first stage of this module is to convert any of these representations into a standard representation for internal use, in the form of an ontology, using a mapping process (Choi et al., 2006; Y. An and Mylopoulos, 2006; Noy and Musen, 2003). The current implementation infers classes and relations from CSV or TSV input files using a rule-based approach, and outputs and ontology in OWL format.

Tagging: This step attaches several tags, such as source, domain, topic and reliability to the mapped ontology. This tags can be either inferred automatically (e.g., the domain and reliability) or provided by the user (e.g., the source). The current implementation requires a manual input by a domain expert.

2.2 Unstructured Data Processing

The sources for unstructured data are extremely varied in format and computational representation. Text is one of the most common forms for storing and communicating human knowledge, but pictures, sound files, and videos are also interesting and increasingly popular forms of communication. Also, in contrast with structured sources, there is a lot of variety in the level of reliability and completeness of unstructured sources.

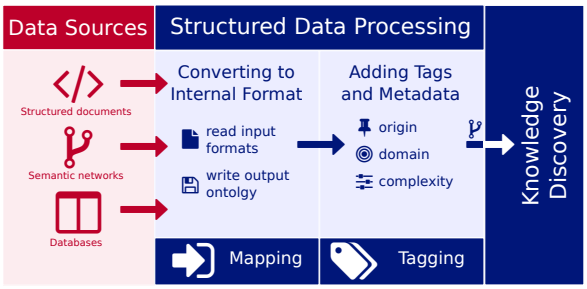

Figure 3: A schema of the Structured Data Module, and a representation of the processes that occur in each of the two main tasks performed by this module.

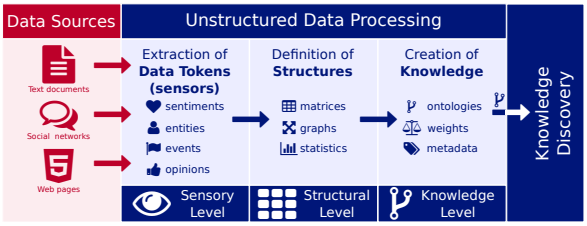

Figure 4: A schema of the three stages of the Unstructured Data Module and its relation with the rest of the framework.

Besides these factors, contrary to structured sources, there is no predefined structure of concepts and relations inside a block of unstructured data. Hence, the module for processing unstructured data is designed as a pipeline through which simple concepts are processed and transformed into more complex ones. Figure 4 shows a schema of this module, as well as an example of the type of processes that occur inside and their relation with each other. The module is organized in a three-levels pipeline as follows:

Sensory Level: Contains a number of processing units called "sensors", which extract different chunks of data. Among the implemented sensors, LETO includes named entity recognition) (Gattani et al., 2013), sentiment analysis (Montoyo et al., 2012), and detection of subject-actions-target triplets (Estevez-Velarde et al., 2018). In general, each of these sensors performs a specific analysis and produces a stream of *data tokens* of a particular type. Each of these data tokens represents a single unit of semantic information, for instance, the existence of a particular entity, or the association between an entity and an event, and are not interrelated.

Structural Level: The data tokens extracted from the original source are processed as a group to find an underlying structure. Techniques implemented in this stage include Latent Semantic Analysis (LSA) (Hofmann, 2017), Principal Components Analysis (PCA) (Guo et al., 2002), Word Embeddings (Turian et al., 2010) and clustering techniques. The output of this stage is either a graph, a correlation matrix, or some statistical description that represents the underlying structure of the data tokens that were previously extracted.

Knowledge Level: The structured information that was previously built is analyzed to refine, remove noise, and extract the relevant pieces of knowledge, based on clustering techniques. This allows synthesizing the knowledge discovered so far according to the context defined by the relations between the semantic units extracted in the previous stage. The output of this stage is always an ontology, which is then passed to the Knowledge Discovery Module for further integration with the stored knowledge. The resulting ontology then becomes part of the stored knowledge of the framework, which is iteratively refined, corrected and enhanced with new knowledge extracted from different sources.

2.3 Knowledge Discovery

The knowledge discovery module receives the output from unstructured data processing and structured data processing, always in the form of an ontology. Each of these ontologies represents a collection of knowledge assets from a particular domain or a general domain. Some of them may overlap, containing the same knowledge facts, even if labeled as different entities or relations. Others may have contradictions or inconsistencies, either within themselves or with one another, see Figure 5. For this purpose, this module performs two main tasks:

Generation: The generation of knowledge involves two processes, namely the merging of ontologies (Noy et al., 2000; Noy and Musen, 2003), and the generation of new (or more general-domain) ontologies from other ontologies (Aussenac-Gilles and Jacques, 2006; Blomqvist, 2009). Merging ontologies requires this module be able to undertake a

matching among entities, relations and instances in two or more ontologies that are deemed similar (Shvaiko and Euzenat, 2013).

Evaluation: After the new ontology is created, this step provides quality evaluation metrics that assert the reliability, completeness or soundness of the new knowledge. These metrics are based on comparing the new ontology with the existing knowledge.

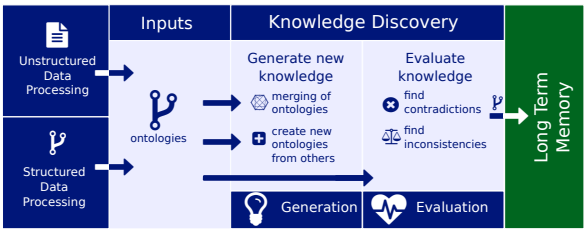

Figure 5: A schema of the architecture of Knowledge Discovery Module, showing the internal tasks performed and its relations with the rest of the framework.

3 Application of LETO to Knowledge Discovery

This section shows the use of the LETO system through a practical scenario that involves the processing of both unstructured and structured data sources. This application illustrates the types of processes (i.e. processing of both unstructured and structured data sources) that our framework performs. We select the Internet Movie Database (IMDB) that contains information about films and actors. We aim to enrich this knowledge with opinions expressed in social networks. Opinions can be extracted from a specific Twitter *hashtag* feed (i.e., `#Oscars`). Figure 6 shows a schematic representation of the whole process.

The first step consist of obtaining the IMDB data (in CSV format) and mapping it to an OWL ontology. Data from IMDB was obtained in tab-separated files, processed by LETO's generic mapping pipeline which infers class names and relation names from the CSV structure. This results in a total of 4,807,262 film instances and 8,427,043 person instances, related by 27,044,985 tuples in 12 different relation types. After the mapping process, the resulting ontology is tagged with relevant metadata. In this case, the domain is **Cinema**, and a high confidence can be assigned since this source is known to be of high quality. These steps are represented in the figure with the numbers *1a* and *1b* and performed in LETO using the *Structured Data Processing* module (see Fig. 3).

The next step involves the processing of a continuous stream of Twitter messages (*2a*). These are obtained through the standard Twitter query API, filtering with the hashtag `#Oscars`, which returned 3375 messages that span a period of 2 weeks. Using standard NLP techniques, each tweet is processed to obtain named entities (Nadeau and Sekine, 2007) and an opinion label (Pang et al., 2008; Liu, 2012) (*2b*). The entity sensor was implemented using spaCy (Honnibal and Montani, 2017), which returned 524 unique PERSON instances, from a total of 1961 PERSON mentions. The document level emotion sensor was implemented through the use of the SAM[1] project (Fernández et al., 2015). An example output of the entity sensor is shown in Figure 7. Similar interfaces are available in LETO for interacting with all the components of the framework, but are not shown for space restrictions.

Afterwards, the different mentions of the same entities across multiple tweets are matched together (*2c*). The least relevant mentions (e.g., those with very few appearances) are filtered out (*2d*). through the clustering technique Affinity Propagation (Pedregosa et al., 2011). Finally, the filtered entities with their associated opinions (*2e*) are tagged and stored in an ontology (*2f*). These steps are performed using components from the *Unstructured Data Processing* module.

After the processing of both structured and unstructured data is completed (*3a*), both sources are selected for a knowledge integration process (*3b*). An ontology mapping technique (Choi et al., 2006) is applied, which maps relevant instances of the IMDB ontologies to their corresponding mentions in the tweets (*3c*). The result of this mapping process is an ontology in the same format as IMDB, but with additional aggregated opinion labels for each instance (of those found in Twitter). This enriched knowledge is tagged (*e3*) and stored for future use. These steps are performed using components from the *Knowledge Discovery* module. The resulting ontology can be visualized in LETO, as shown in Figure 9. This visualization tool shows both the classes and instances, enabling an interactive exploration of the ontology.

[1] http://wiki.socialisingaroundmedia.com/

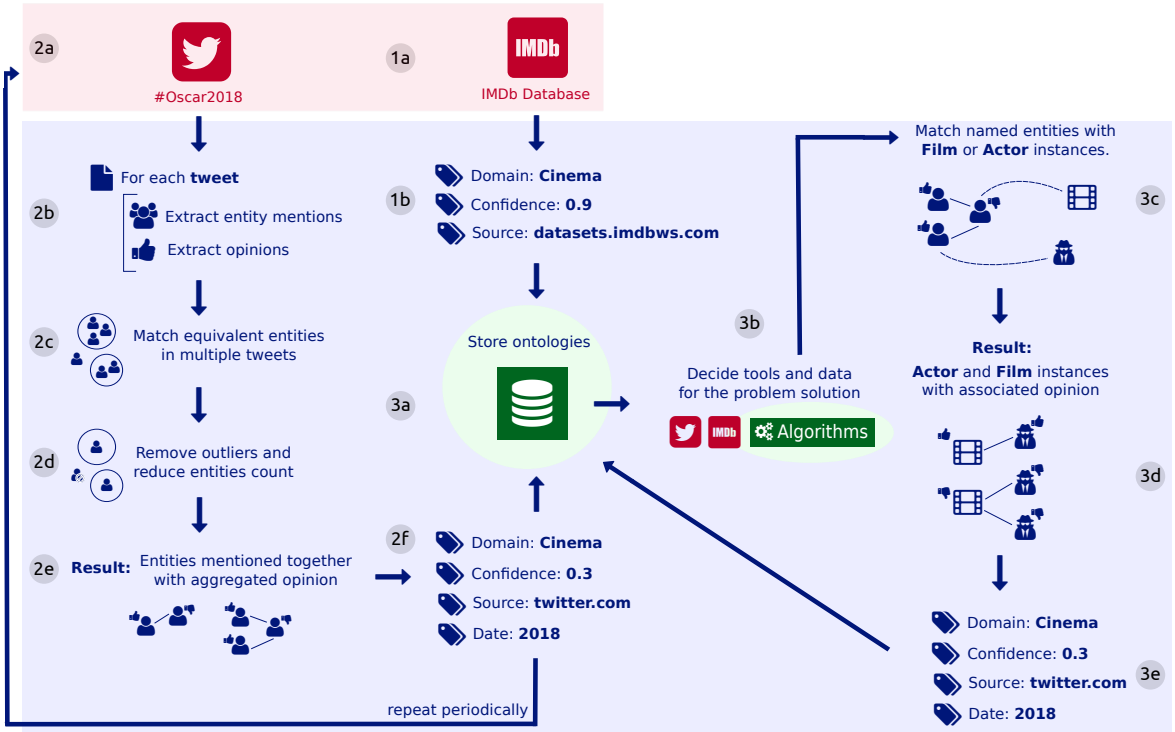

Figure 6: Schematic representation of the process for mapping emotions in reaction to movie reviews with IMDB.

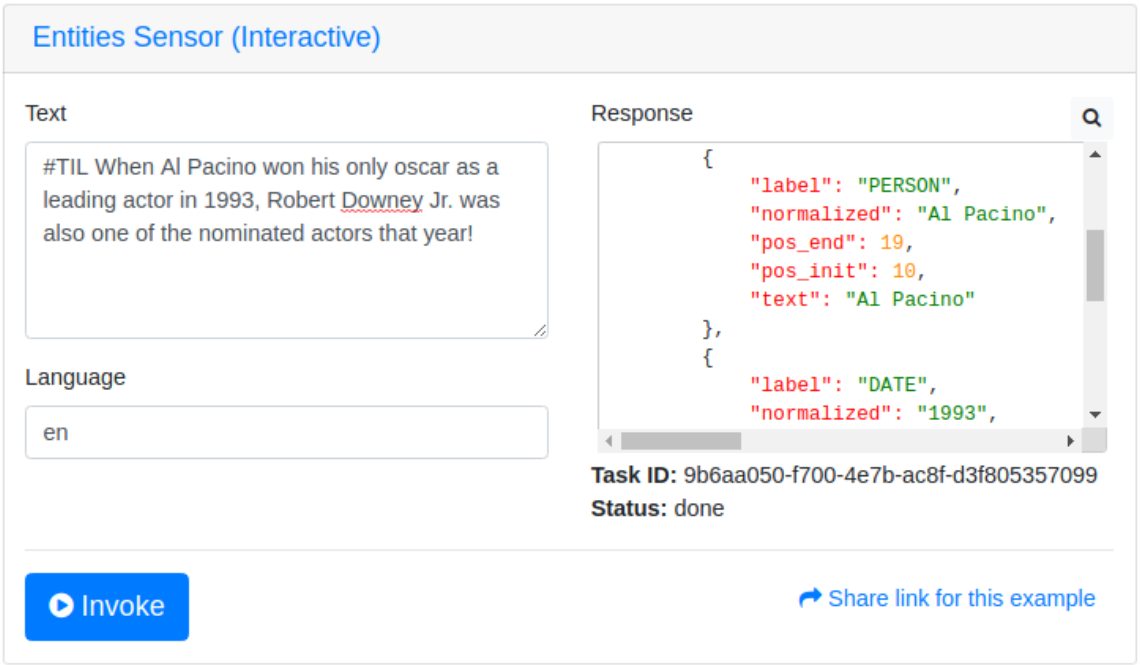

Figure 7: Example execution of the Entity Sensor for one tweet. A similar interface allows the batch execution for a collection of tweets.

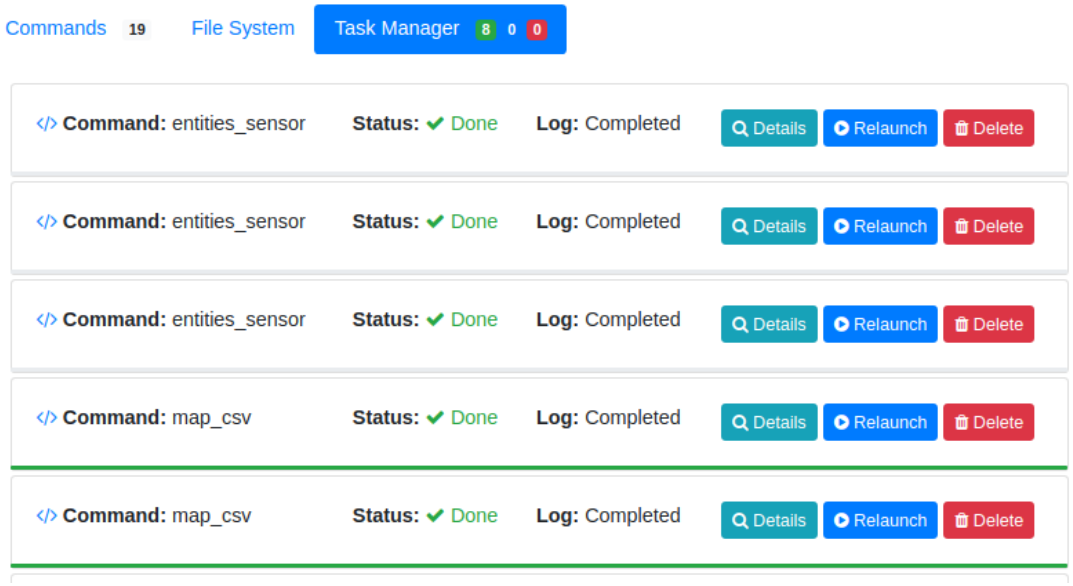

Figure 8: Main UI of LETO, specifically the Task Management view.

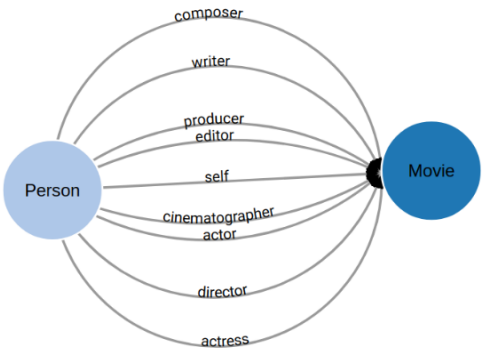

Figure 9: Visualizing the structure of the IMDB ontology in LETO.

The knowledge generation process involved matching Twitter PERSON instances with IMDB instances and attaching an average of the emotions found in each mention of the corresponding instance. A total of 212 instances were matched, which indicates a 40.45% of accuracy for the Twitter entity extractor. A manual review of the 542 recognized instances was performed, to evaluate the reasons for the mistakes. All entities appearing in Twitter where searched in Google and the first result was used as ground truth. Table 1 summarizes these results.

The current implementation of LETO provides an interactive application where researchers can apply the different algorithms and techniques implemented in each module, both interactively (i.e., using a single input example) or in batch mode.

Metric	Value	Percent
Correct matches	212	**40.45**
Correct mismatch	19	**3.62**
Matching error	118	22.52
Extraction error	165	31.48
Knowledge error	10	1.91
Context missing	2	0.38
Total errors	293	55.92

Table 1: Summary of results of the knowledge discovery process.

LETO supports multiple processes running in parallel, and provides tools for running and monitor long-term processes that can take hours or days. Figure 8 shows a overall view of LETO's main user interface, specifically the view for task management.

4 Conclusion and Future Works

In this research work, the aim was to design and implement a framework for automatic knowledge discovery from different data sources. We considered the discovery of knowledge from structured and unstructured sources of information. This framework has been designed as a modular set of components that perform specific tasks and communicate with each other. An open-source prototype implementation of LETO is currently available[2], which already contains several of the main components. In future lines of development, we will pursue the implementation of more var-

[2]https://github.com/knowledge-learning/leto

ied sensors, and more complex mechanisms for knowledge integration (e.g., ontology merging and mapping processes). Another line for future research is related to context mismatch and recognition, specifically in the *Unsupervised Processing Module*. This process is necessary for accurately matching portions of unstructured text to sections of an already stored ontology. We will also focus on extending the automation processes currently available in LETO.

Acknowledgments

This research has been supported by a Carolina Foundation grant in agreement with University of Alicante and University of Havana. Moreover, it has also been partially funded by both aforementioned universities, the Generalitat Valenciana and the Spanish Government through the projects SIIA (PROMETEU/2018/089), LIVING-LANG (RTI2018-094653-B-C22) and INTEGER (RTI2018-094649-B-I00).

References

Harith Alani, Sanghee Kim, David E Millard, Mark J Weal, Wendy Hall, Paul H Lewis, and Nigel R Shadbolt. 2003. Automatic ontology-based knowledge extraction from web documents. *IEEE Intelligent Systems* 18(1):14–21.

Nathalie Aussenac-Gilles and Marie-Paule Jacques. 2006. Designing and evaluating patterns for ontology enrichment from texts. In *International Conference on Knowledge Engineering and Knowledge Management*. Springer, pages 158–165.

Ken Barker, Bhalchandra Agashe, Shaw Yi Chaw, James Fan, Noah Friedland, Michael Glass, Jerry Hobbs, Eduard Hovy, David Israel, Doo Soon Kim, et al. 2007. Learning by reading: A prototype system, performance baseline and lessons learned. In *AAAI*. volume 7, pages 280–286.

Eva Blomqvist. 2009. Ontocase-automatic ontology enrichment based on ontology design patterns. In *International Semantic Web Conference*. Springer, pages 65–80.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems* 30(1-7):107–117.

Paul Buitelaar, Philipp Cimiano, Stefania Racioppa, and Melanie Siegel. 2006. Ontology-based information extraction with soba. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

Paul Buitelaar and Michael Sintek. 2004. Ontolt version 1.0: Middleware for ontology extraction from text. In *Proc. of the Demo Session at the International Semantic Web Conference*.

Silvana Castano, Sofia Espinosa, Alfio Ferrara, Vangelis Karkaletsis, Atila Kaya, Sylvia Melzer, Ralf Möller, Stefano Montanelli, and Georgios Petasis. 2007. Ontology dynamics with multimedia information: The boemie evolution methodology. In *International Workshop on Ontology Dynamics (IWOD-07)*. page 41.

Balakrishnan Chandrasekaran. 1986. Generic tasks in knowledge-based reasoning: High-level building blocks for expert system design. *IEEE expert* 1(3):23–30.

Namyoun Choi, Il-Yeol Song, and Hyoil Han. 2006. A survey on ontology mapping. *ACM Sigmod Record* 35(3):34–41.

Philipp Cimiano, Alexander Mädche, Steffen Staab, and Johanna Völker. 2009. Ontology learning. In *Handbook on ontologies*, Springer, pages 245–267.

Philipp Cimiano and Johanna Völker. 2005. text2onto. In *International Conference on Application of Natural Language to Information Systems*. Springer, pages 227–238.

Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom Mitchell, Kamal Nigam, and Seán Slattery. 2000. Learning to construct knowledge bases from the world wide web. *Artificial intelligence* 118(1-2):69–113.

James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. 2010. The youtube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, pages 293–296.

Euthymios Drymonas, Kalliopi Zervanou, and Euripides GM Petrakis. 2010. Unsupervised ontology acquisition from plain texts: the OntoGain system. In *International Conference on Application of Natural Language to Information Systems*. Springer, pages 277–287.

S. Estevez-Velarde, Y. Gutierrez, A. Montoyo, A. Piad-Morffis, R. Munoz, and Y. Almeida-Cruz. 2018. Gathering object interactions as semantic knowledge (accepted). In *Proceedings of the 2017 International Conference on Artificial Intelligence (ICAI'17)*.

David Faure and Thierry Poibeau. 2000. First experiments of using semantic knowledge learned by asium for information extraction task using intex. In *Proceedings of the ECAI workshop on Ontology Learning*.

Javi Fernández, Yoan Gutiérrez, José M Gómez, and Patricio Martínez-Barco. 2015. Social rankings: análisis visual de sentimientos en redes sociales. *Procesamiento del Lenguaje Natural* 55:199–202.

David Ferrucci, Anthony Levas, Sugato Bagchi, David Gondek, and Erik T Mueller. 2013. Watson: beyond jeopardy! *Artificial Intelligence* 199:93–105.

Abhishek Gattani, Digvijay S Lamba, Nikesh Garera, Mitul Tiwari, Xiaoyong Chai, Sanjib Das, Sri Subramaniam, Anand Rajaraman, Venky Harinarayan, and AnHai Doan. 2013. Entity extraction, linking, classification, and tagging for social media: a wikipedia-based approach. *Proceedings of the VLDB Endowment* 6(11):1126–1137.

Richard Gross. 2015. *Psychology: The science of mind and behaviour 7th edition*. Hodder Education.

Q Guo, W Wu, DL Massart, C Boucon, and S De Jong. 2002. Feature selection in principal component analysis of analytical data. *Chemometrics and Intelligent Laboratory Systems* 61(1-2):123–132.

Jesús M Hermida, Santiago Meliá, Jose-Javier Martínez, Andrés Montoyo, and Jaime Gómez. 2012. Developing semantic rich internet applications with the s m 4ria extension for oide. In *International Conference on Web Engineering*. Springer, pages 20–25.

Thomas Hofmann. 2017. Probabilistic latent semantic indexing. In *ACM SIGIR Forum*. ACM, volume 51, pages 211–218.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear* .

Murphy Kevin. 2012. Machine learning: a probabilistic perspective.

Sang-Soo Kim, Jeong-Woo Son, Seong-Bae Park, Se-Young Park, Changki Lee, Ji-Hyun Wang, Myung-Gil Jang, and Hyung-Geun Park. 2008. Optima: An ontology population system. In *3rd Workshop on Ontology Learning and Population (July 2008)*.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5(1):1–167.

T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2018. Never-ending learning. *Commun. ACM* 61(5):103–115. https://doi.org/10.1145/3191513.

Andrés Montoyo, Patricio MartíNez-Barco, and Alexandra Balahur. 2012. Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes* 30(1):3–26.

Natalya F Noy and Mark A Musen. 2003. The prompt suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies* 59(6):983–1024.

Natalya Fridman Noy, Mark A Musen, et al. 2000. Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-00). Available as SMI technical report SMI-2000-0831*.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval* 2(1–2):1–135.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Georgios Petasis, Vangelis Karkaletsis, Georgios Paliouras, Anastasia Krithara, and Elias Zavitsanos. 2011. Ontology population and enrichment: State of the art. In *Knowledge-driven multimedia information extraction and ontology evolution*. Springer-Verlag, pages 134–166.

Pavel Shvaiko and Jérôme Euzenat. 2013. Ontology matching: state of the art and future challenges. *IEEE Transactions on knowledge and data engineering* 25(1):158–176.

Fabian M Suchanek, Georgiana Ifrim, and Gerhard Weikum. 2006. Leila: Learning to extract information by linguistic analysis. In *Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*. pages 18–25.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, pages 384–394.

Panos Vassiliadis. 2009. A survey of extract–transform–load technology. *International Journal of Data Warehousing and Mining (IJDWM)* 5(3):1–27.

Nicolas Weber and Paul Buitelaar. 2006. Web-based ontology learning with isolde. In *Proc. of the Workshop on Web Content Mining with Human Language at the International Semantic Web Conference, Athens GA, USA*. volume 11.

A. Borgida Y. An and J. Mylopoulos. 2006. Building semantic mappings from databases to ontologies. volume 21st National Conference on Artificial Intelligence (AAAI 06).