Introducing Distiller: a Lightweight Framework for Knowledge Extraction and Filtering

Dario De Nart, Dante Degl'Innocenti, Carlo Tasso

Artificial Intelligence Lab
Department of Mathematics and Computer Science
University of Udine, Italy
{dario.denart,carlo.tasso}@uniud.it, dante.deglinnocenti@spes.uniud.it

Abstract. Semantic content analysis is an activity that can greatly support a broad range of user modelling applications. Several automatic tools are available, however such systems usually provide little tuning possibilities and do not support integration with different systems. Personalization applications, on the other hand, are becoming increasingly multi-lingual and cross-domain. In this paper we present a novel framework for Knowledge Extraction, whose main goal is to support the development of new strategies and technologies and to ease the integration of the existing ones.

1 Introduction

Adaptive Personalization systems can greatly benefit from automatic Knowledge Extraction (herein KE) from natural language text: for instance machine readable data about the textual content of visited Web pages or of user-generated content can enhance user profiling activities. Due to the current size of the Web, however, one cannot expect human experts to annotate such data manually. Several tools have been developed over the past years to address this issue. However we can identify three critical issues in state-of-the-art Knowledge Extraction systems:

- *Multilinguality*: roughly half of the available Web pages include non-English text [1]. The large majority of Web users are non-English native speakers, and multilingual personalization is a hot research topic, however KE tools show a general lack of multilingual non-English support.
- *Knowledge Source Completeness*: KE systems mostly rely on a specific knowledge source (such as DBpedia) acting in a closed-world fashion and assuming that such knowledge source is complete. This assumption is in contrast with the open-world assumption of semantic Web technologies and shows off its limitations when applied to texts such as scientific papers, where new concepts are often introduced. Therefore a more flexible approach open to more than one external knowledge source seems more appropriate.

[1] http://w3techs.com/technologies/overview/content_language/all

- *Knowledge Overload*: long texts, such as scientific papers, may include a lot of named entities, but not all are equally relevant inside the text. State-of-the-art KE systems currently provide Named Entity Recognition, but do not filter relevant entities nor include relevance measures.

In this paper we introduce *Distiller*, a KE framework whose aim is to overcome these limitations and allow integration of heterogeneous KE technologies.

2 Related Work

Named entity recognition and automatic semantic data generation from natural language text has already been investigated and several knowledge extraction systems already exist [4], such as OpenCalais [2], Apache Stanbol[3], and TagMe [3]. In [8] an ensemble learning strategy to raise the accuracy of the named entity identification process is presented. Several authors in the literature have addressed the problem of filtering document information by identifying keyphrases (herein KPs) and a wide range of approaches have been proposed. The authors of [10] identify four types of KP extraction strategies:

- *Simple Statistical Approaches*: mostly unsupervised techniques, considering word frequency, TF-IDF or word co-occurency [7].
- *Linguistic Approaches*: techniques relying on linguistic knowledge to identify KPs. Proposed methods include lexical analysis, syntactic analysis, and discourse analysis [5].
- *Machine Learning Approaches*: techniques based on machine learning algorithms such as Naive Bayes classifiers and SVM. Systems such as KEA [9] belong to this category.
- *Other Approaches*: other strategies exist which do not fit into one of the above categories, mostly hybrid approaches combining two or more of the above techniques [1]. Among others, heuristic approaches based on knowledge-based criteria [6] have been proposed.

3 System Overview

In order to overcome the shortcomings of state-of-the-art KE systems we extended the approach presented in [2] and formalized it as a framework named *Distiller* whose main aim is to support research and prototyping activities by providing an environment for building testbed systems and integrating existing systems. The guiding principle of the framework design is that several different types of knowledge are involved in the process of KE and should be clearly separated to design systems able to cope with multilinguality and multi-domain issues. We consider four main types of knowledge: Statistical, Linguistic, External (i.e. coming from outside the text, like the one extracted from ontologies),

[2] http://www.opencalais.com/

[3] https://stanbol.apache.org/

and Heuristic knowledge. Linguistic knowledge is language dependant, Heuristic knowledge is domain dependent, and External knowledge is both domain and language dependant. At a more practical level, this principle implies that different types of knowledge must reside in distinct modules, for instance, statistical and linguistic analysis must be handled by different modules.

Distiller is organized in a series of single-knowledge oriented modules and its workflow is organized in four phases: Concept Unit Splitting, Annotation, Candidate Generation, and Filtering, as shown in Figure 1. In the first phase the text is split into *Concept Units*, i.e. logical blocks such as chapters, paragraphs or sentences. The framework allows the co-existence of concept units of different languages inside a document. The Annotation phase consists in enriching the text with information such as POS tagging, stems, lemmas, or links to entities from external knowledge sources (such as DBpedia). This phase introduces new knowledge in the text, and several different annotators can contribute, enriching the text with different kinds of knowledge, but mostly with External knowledge that may come from heterogeneous sources. Existing KE tools, such as TagMe, can be integrated in the framework as annotators. The Candidate Generation phase identifies in the text all the candidate entities and/or concepts of interests exploiting the annotations provided in the previous step and internally represents them as KPs with an attached set of annotations. Finally, the Filtering phase evaluates a relevance score for each candidate concept depending on which it is returned as output or hidden. The Filtering phase, like the Candidate Generation one, may exploit different types of knowledge embedded in annotations, and combine them according to the needs of the applications that will eventually use the extracted knowledge.

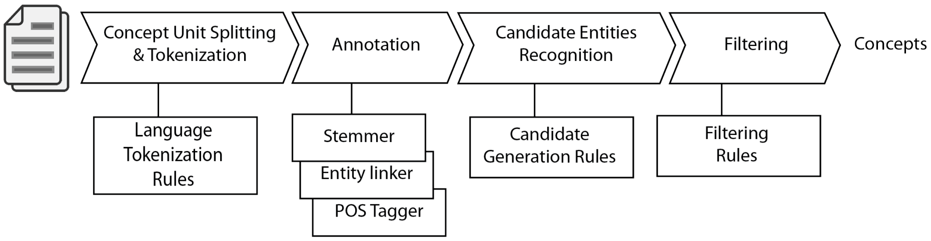
Concept Unit Splitting & Tokenization
Annotation
Candidate Entities Recognition
Filtering
Concepts
Language Tokenization Rules
Stemmer
Entity linker
POS Tagger
Candidate Generation Rules
Filtering Rules

Fig. 1: Framework workflow.

Distiller is implemented in Java using the Dependency Injection pattern, that allows users to easily switch between different modules and configurations. Default implementations for all the above described modules are provided with the framework[4].

[4] A sample application built with the default modules is showcased at ailab.uniud.it:8080/distiller

4 Conclusions

With respect to the three issues of KE presented in Section 1, Distiller allows the development of applications able to overcome such shortcomings. The issue of multilinguality is eased by the possibility of specifying a wide array of annotators and to dynamically link them at runtime depending on the text language. The issue of Knowledge Source Completeness is eased by the possibility of integrating heterogeneous knowledge sources as different annotators and implementing annotators who generate URIs on the fly. The issue of Knowledge Overload, finally, is eased by the presence of a filtering phase in which entities are evaluated with respect to their relevance in the text.

References

1. De Nart, D., Tasso, C.: A domain independent double layered approach to keyphrase generation. In: WEBIST 2014 - Proceedings of the 10th International Conference on Web Information Systems and Technologies. pp. 305–312. SCITEPRESS Science and Technology Publications (2014)
2. Degl'Innocenti, D., De Nart, D., Tasso, C.: A new multi-lingual knowledge-base approach to keyphrase extraction for the italian language. In: Proceedings of the 6th International Conference on Knowledge Discovery and Information Retrieval. pp. 78–85. SciTePress (2014)
3. Ferragina, P., Scaiella, U.: Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management. pp. 1625–1628. CIKM '10, ACM, New York, NY, USA (2010)
4. Gangemi, A.: A comparison of knowledge extraction tools for the semantic web. In: The Semantic Web: Semantics and Big Data, pp. 351–366. Springer (2013)
5. Krapivin, M., Marchese, M., Yadrantsau, A., Liang, Y.: Unsupervised key-phrases extraction from scientific papers using domain and linguistic knowledge. In: Digital Information Management, 2008. ICDIM 2008. Third International Conference on. pp. 105–112 (Nov 2008)
6. Liu, Z., Li, P., Zheng, Y., Sun, M.: Clustering to find exemplar terms for keyphrase extraction. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1. pp. 257–266. EMNLP '09 (2009)
7. Matsuo, Y., Ishizuka, M.: Keyword extraction from a single document using word co-occurrence statistical information. International Journal on Artificial Intelligence Tools 13(01), 157–169 (2004)
8. Speck, R., Ngonga Ngomo, A.C.: Ensemble learning for named entity recognition. In: The Semantic Web ISWC 2014, Lecture Notes in Computer Science, vol. 8796, pp. 519–534. Springer International Publishing (2014)
9. Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., Nevill-Manning, C.G.: Kea: Practical automatic keyphrase extraction. In: Proceedings of the fourth ACM conference on Digital libraries. pp. 254–255. ACM (1999)
10. Zhang, C.: Automatic keyword extraction from documents using conditional random fields. Journal of Computational Information Systems 4(3), 1169–1180 (2008), http://eprints.rclis.org/handle/10760/12305