The Distiller Framework: Current State and Future Challenges

Marco Basaldella(✉), Giuseppe Serra, and Carlo Tasso

Laboratorio di Intelligenza Artificiale, Università degli Studi di Udine,
Via delle Scienze 208, Udine, Italy
{marco.basaldella,giuseppe.serra,carlo.tasso}@uniud.it

Abstract. In 2015, we introduced a novel knowledge extraction framework called the Distiller Framework, with the goal of offering the research community a flexible, multilingual information extraction framework [3]. Two years later, the project has significantly evolved, by supporting more languages and many machine learning algorithms. In this paper we present the current design of the framework and some of its applications.

Keywords: Information extraction · Keyphrase extraction
Named entity recognition

1 Introduction

Today digital document archives contain a tremendous amount of documents of various types, such as books, articles, papers, reports etc. Therefore, there is a urgent demand for adequate tools to semantically process documents in order to support the user needs. Based on this demand, in this paper we present the current state of the Distiller framework, an open source information extraction framework developed in the Artificial Intelligence Laboratory of the University of Udine. The Distiller framework allows annotating any document with linguistic, statistical, semantic or any kind of information.

We present the history of the framework and related research in Sect. 2; in Sect. 3, we describe the design of the framework; then, in in Sect. 4, we explain how to download and run the Distiller. In Sect. 5, we briefly present research performed using the Distiller framework in the fields of Keyphrase Extraction and Named Entity Recognition in the biomedical domain. Finally, Sect. 6 presents the challenges that we will have to face in the future for continuing the development of the framework.

2 Related Work

The roots of the Distiller framework are in the Automatic Keyphrase Extraction (herein AKE) system DIKpE [19]. Originally, the system was part of a content recommendation framework, and performed AKE using five features and

G. Serra and C. Tasso (Eds.): IRCDL 2018, CCIS 806, pp. 93–100, 2018.
https://doi.org/10.1007/978-3-319-73165-0_9

heuristically selected weights. Later, [10,11] extended the approach, offering the possibility of inferring keyphrases not contained in the original document and of processing documents in Italian as well. However, the software used in these projects was adapted using a series of ad-hoc solutions, hence becoming difficult to maintain and to further extend with new functionality. For these reasons, we introduced the Distiller framework in [3], with the goal of building a more maintainable system which could be also used for tasks different than AKE.

Other open source KE systems exist in academia. KEA [24], one of the first AKE algorithms, is available online as open source software[1], but the project seems abandoned since 2007. A free implementation of the RAKE [21] algorithm is available online as well[2], but with little or no possible customization. PKE [8] is an open source[3] implementation of many KE algorithms, such as KEA, TopicRank [9], WINGNUS [16] and others. However, it is focused on keyphrase extraction only and it cannot be used for other NLP tasks. The MAUI software[4] seems the closest system to the Distiller framework, offering an open source implementation of an improved version of the KEA algorithm and algorithms for Named Entity Recognition or Automatic Tagging [15]. However, many of these features are only available buying a commercial license, and the end user is left with no or little possibility of customizing the pipelines.

3 Design

The Distiller framework has been developed in Java 8, due to the robustness of the language, its strong object-oriented paradigm, and due to the availability of a large number of NLP and machine learning tools already available for this language, such as the Stanford CoreNLP library [14], Apache OpenNLP[5], Weka [22], and others. Moreover, Java gives the possibility of writing wrappers to other software, for added flexibility. Many already available wrappers are developed by the open source community, like e.g. for generic tools like R or Matlab, or for specialized tools like e.g. CRFSuite [17].

The design of the framework is somewhat similar to the Stanford CoreNLP system [14]. In fact, like in Stanford CoreNLP, we offer the possibility to annotate the text with a sequence of `Annotator` objects. When the developer of an information extraction pipeline is working with the Distiller, he will mainly work using the following classes of the framework:

`DocumentComponent`: this class represents a unit of information within a document. Such unit may be a chapter, a paragraph, a sentence, or just the whole document. It is designed using the Composite pattern [12], where the composite object (sentence, chapter, section...) is represented by the

[1] http://www.nzdl.org/Kea/download.html.
[2] https://github.com/aneesha/RAKE.
[3] https://github.com/boudinfl/pke.
[4] https://github.com/zelandiya/maui.
[5] http://opennlp.apache.org.

`DocumentComposite` class and the smallest component is represented by the `Sentence` class, which is in turn an aggregation of `Token`s.

`Blackboard`: this is the class that contains the original document and that will contain all the information produced by the pipeline. It consists of a pointer to the root `DocumentComponent` of the document and a dictionary of `Annotation`s that can be filled at according to the specific application considered.

`Annotation`: the class that represents an annotation. It can be added to the Blackboard or any `Annotable` object. Example of `Annotable` objects are any `DocumentComponent`, `Token`s, `Gram`s, etc.

`Annotator`: an `abstract` class that has to be extended by any class that produces `Annotation`s. An `Annotator` can be a part-of-speech tagger, it can count the occurrences of a word in a document, it can call an external knowledge base (e.g. Wikipedia) to get more information, it can be a machine learning algorithm, and so on.

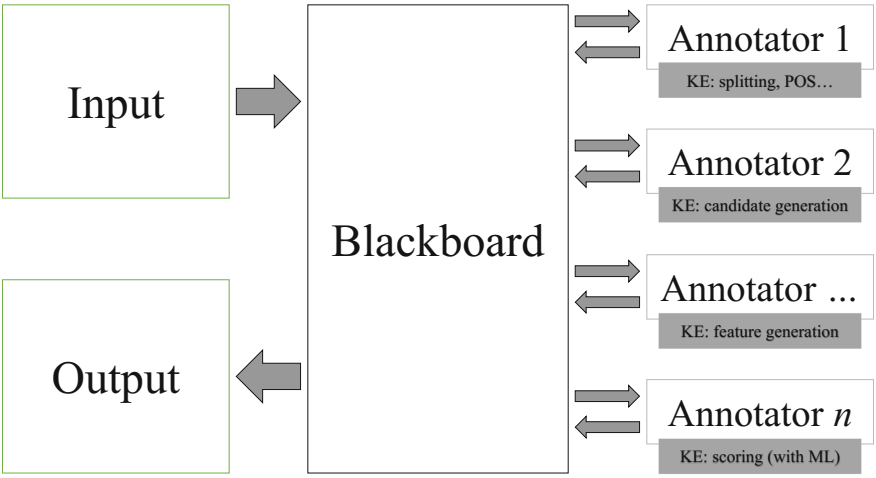

Fig. 1. The high-level architecture of the framework. The workflow is the following: first, the document is written on the blackboard. Then, a sequence of `Annotator`s annotate the document, eventually using previously produced annotations. When all annotators finish their job, the produced annotations are returned as output. In this case, we put some example annotators used for Keyphrase Extraction.

Figure 1 shows an example workflow using the Distiller framework, applied to the case of the Keyphrase Extraction task. In this case, the annotators contained in the pipeline will perform the following operations:

1. **Language detection:** first, they detect the language of the document;
2. **Low-level NLP:** then, they perform low-level NLP operations on the document, such as tokenization, part-of-speech tagging, stemming, and so on;

3. **Candidate Generation:** using the information produced in the previous step, they generate the candidate keyphrases that match certain part-of-speech patterns [2,19];
4. **Candidate Annotation:** they annotate the candidate keyphrases with information from different knowledge domains, such as statistics (e.g. number of occurrences of the candidate, length of the candidate, ...), linguistics (number of nouns in the candidate [19], anaphors that have the candidate as antecedent [2], ...), or from external knowledge (e.g. Wikipedia [3]), and so on;
5. **Candidate Scoring:** they score the candidates using the annotations produced in the previous steps. The score can be calculated using simple, handcrafted techniques [3,19] or using machine learning algorithms [2].

Some annotators are already provided out-of-the-box. For example, we provide two wrappers for Stanford CoreNLP and Apache OpenNLP that offer sentence segmentation, word tokenization, and part-of-speech tagging in many languages (see Sect. 5), a wrapper for the Porter's stemmer algorithm [18], a module that calculates statistical information about n-grams contained in the document, and so on.

4 Obtaining and Running the Distiller Framework

Distiller is available as an open source project under the GPLv2 license, and it is available online at https://github.com/ailab-uniud/distiller-CORE.

After building it with Maven, it is possible to run the keyphrase extraction pipeline described in Sect. 3 by writing the following code:

```
String document = ... // load the input document
Distiller d = distiller = DistillerFactory.
    loadFromPackagedXML(``pipelines/defaultKE.xml");
Blacboard b = d.distill(document);
Collection<Keyphrase> keyphrase =
    b.getGramsByType(Keyphrase.KEYPHRASE);
```

This code will load the default keyphrase extraction pipeline, run it, and store the results in the `keyphrase` variable. Please note that the `defaultKE` pipeline requires R installed on the system; alternatively, one can run the implementation of [19], called `fastKE`, which does not need any additional software.

5 Applications

Since the beginning of the development of the framework we immediately started to use it in actual research tasks, in order to gain experience about the challenges that developers face in designing tools for the academic world. We actually claim that the flexibility of the Distiller framework, together with its easy customizability, make it an ideal testbed for exploration and for R&D activities.

The following list summarizes the mayor applications and research activities carried out so far in the Artificial Intelligence Laboratory of the University of Udine with the Distiller framework.

Putting More Linguistic and Keyphrase Extraction

We successfully used the Distiller framework to demonstrate the possibility of extracting better keyphrases using more linguistic knowledge than in the classic statistics based approaches [2]. In particular, we exploited the field of Anaphora Resolution (herein AR), obtaining an improvement in performance when adding AR-based information to the KE task. To obtain this result, we used the AR capabilities of the Stanford CoreNLP library to develop two pipelines: one pipeline that replaced anaphors with their antecedents, and one that used AR-based `Annotator`s and `Annotation`s to score the keyphrases.

Multilinguality and Keyphrase Extraction

We implemented a five-language keyphrase extraction pipeline, capable to process documents in English, Arabic, Portuguese, Romanian and Italian [5]. However, we had training data only for English and Arabic. Thus, to prove the effectiveness of our approach, we trained a machine learning algorithm over the two languages for which we had training data and tested it on custom collected datasets in all five languages. The results show that, even if trained and tested over different languages, the statistical approach of keyphrase extraction is still effective.

Entity Recognition in the Biomedical Domain

We used the Distiller for entity recognition and linking in the biomedical domain as well [4], demonstrating its flexibility. In this work, Distiller was used along OntoGene [20], a text mining framework developed by the University of Zurich, in order to build an hybrid dictionary based - machine learning system for detection and linking of technical terms in the biomedical domain. The system, based on Conditional Random Fields, obtained promising results on the CRAFT corpus, with increased F1-Score when compared to the current state-of-the art systems.

6 Conclusions and Future Work

In the last years, Deep Learning (abbrv. DL) techniques are attacking "classical" machine learning approaches in many fields, outperforming them in many tasks. For example, in the Machine Translation domain, the WMT 2016 task saw a surge of Neural Machine Translation systems, which vastly outperformed the syntax-based systems presented in the previous edition [6,7]. The same happened in the ImageNet competition, where the introduction of DL techniques

brought the error rate down to 3,6% from the previous, pre-"Deep Learning era" 26,1% state-of-the art [13]. DL approaches also improved the state of the art in many other fields, such as speech recognition and image segmentation, and are currently regarded of obtaining "superhuman" performance in traffic sign classification [13].

DL techniques have been developed also for AKE [25], Named Entity Recognition [23], and many other NLP tasks with very promising results. This will prove a challenge for systems designed to be knowledge-based like the Distiller framework. However, there are tasks where the "classic" knowledge-based approach is still dominant, as NER and Concept Recognition in specialized fields, due to the need of binding concepts to specialized ontologies [4]. In these cases, where algorithms need to take into account rare words that DL models often fail to recognize, we believe systems like the Distiller still has much to offer to the research community. In addition we believe that, due to the extensible framework architecture of Distiller, our system will continue to be useful in the future, e.g. by integrating deep learning libraries inside it. For example, the popular Tensorflow [1] library offers Java APIs, so it would be easy to integrate it in our system. In the future, our goal is to use the Distiller framework to develop an hybrid approach for AKE, which takes into account both the "classic" supervised features, along with the new, DL based techniques.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015). http://tensorflow.org/
2. Basaldella, M., Chiaradia, G., Tasso, C.: Evaluating anaphora and coreference resolution to improve automatic keyphrase extraction. In: COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, Osaka, Japan, 11–16 December 2016, pp. 804–814 (2016)
3. Basaldella, M., De Nart, D., Tasso, C.: Introducing distiller: a unifying framework for knowledge extraction. In: Proceedings of 1st AI*IA Workshop on Intelligent Techniques At Libraries and Archives co-located with XIV Conference of the Italian Association for Artificial Intelligence (AI*IA 2015). Associazione Italiana per l'Intelligenza Artificiale (2015)
4. Basaldella, M., Furrer, L., Colic, N., Ellendorff, T., Tasso, C., Rinaldi, F.: Using a hybrid approach for entity recognition in the biomedical domain. In: Proceedings of the 7th International Symposium on Semantic Mining in Biomedicine, SMBM 2016, Potsdam, Germany, 4–5 August 2016, pp. 11–19 (2016)
5. Basaldella, M., Helmy, M., Antolli, E., Popescu, M.H., Serra, G., Tasso, C.: Exploiting and evaluating a supervised, multilanguage keyphrase extraction pipeline for under-resourced languages. In: Recent Advances in Natural Language Processing 2017 (RANLP 2017), Varna (Bulgaria), 4–6 September 2017

6. Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Jimeno Yepes, A., Koehn, P., Logacheva, V., Monz, C., Negri, M., Neveol, A., Neves, M., Popel, M., Post, M., Rubino, R., Scarton, C., Specia, L., Turchi, M., Verspoor, K., Zampieri, M.: Findings of the 2016 conference on machine translation. In: Proceedings of the First Conference on Machine Translation, pp. 131–198. Association for Computational Linguistics, Berlin, August 2016
7. Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Huck, M., Hokamp, C., Koehn, P., Logacheva, V., Monz, C., Negri, M., Post, M., Scarton, C., Specia, L., Turchi, M.: Findings of the 2015 workshop on statistical machine translation. In: Proceedings of the Tenth Workshop on Statistical Machine Translation, pp. 1–46. Association for Computational Linguistics, Lisbon, September 2015
8. Boudin, F.: pke: an open source python-based keyphrase extraction toolkit. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations. pp. 69–73. The COLING 2016 Organizing Committee, Osaka, December 2016
9. Bougouin, A., Boudin, F., Daille, B.: Topicrank: graph-based topic ranking for keyphrase extraction. In: Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, 14–18 October 2013, pp. 543–551 (2013)
10. 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 (2014)
11. 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)
12. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-oriented Software. Addison-Wesley Longman Publishing Co., Inc., Boston (1995)
13. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016). http://www.deeplearningbook.org
14. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations, pp. 55–60 (2014)
15. Medelyan, O., Frank, E., Witten, I.H.: Human-competitive tagging using automatic keyphrase extraction. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, vol. 3, pp. 1318–1327. Association for Computational Linguistics, Stroudsburg (2009)
16. Nguyen, T.D., Luong, M.: WINGNUS: keyphrase extraction utilizing document logical structure. In: Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, 15–16 July 2010, pp. 166–169 (2010). http://aclweb.org/anthology/S/S10/S10-1035.pdf
17. Okazaki, N.: Crfsuite: a fast implementation of conditional random fields (crfs) (2007). http://www.chokkan.org/software/crfsuite/
18. Porter, M.F.: An Algorithm for suffix stripping. In: Readings in Information Retrieval, pp. 313–316. Morgan Kaufmann Publishers Inc., San Francisco (1997)
19. Pudota, N., Dattolo, A., Baruzzo, A., Ferrara, F., Tasso, C.: Automatic keyphrase extraction and ontology mining for content-based tag recommendation. Int. J. Intell. Syst. **25**(12), 1158–1186 (2010)

20. Rinaldi, F.: The ontogene system: an advanced information extraction application for biological literature. EMBnet.journal **18**(B) (2012)
21. Rose, S., Engel, D., Cramer, N., Cowley, W.: Automatic keyword extraction from individual documents. In: Text Mining, pp. 1–20 (2010)
22. Russell, I., Markov, Z.: An introduction to the weka data mining system (abstract only). In: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, Seattle, WA, USA, 8–11 March 2017, p. 742 (2017)
23. dos Santos, C., Guimarães, V.: Boosting named entity recognition with neural character embeddings. In: Proceedings of the Fifth Named Entity Workshop, pp. 25–33. Association for Computational Linguistics, Beijing, July 2015
24. 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)
25. Zhang, Q., Wang, Y., Gong, Y., Huang, X.: Keyphrase extraction using deep recurrent neural networks on Twitter. In: Proceedings of Conference on Empirical Methods in Natural Language Processing (2016)