

Digging in the Dirt: Extracting Keyphrases from Texts with KD

Giovanni Moretti¹, Rachele Sprugnoli¹⁻², Sara Tonelli¹

¹Fondazione Bruno Kessler, Trento

²Università di Trento

{moretti, sprugnoli, satonelli}@fbk.eu

Abstract

English. In this paper we present a keyphrase extraction system called Keyphrase Digger (KD). The tool uses both statistical measures and linguistic information to detect a weighted list of n-grams representing the most important concepts of a text. KD is the reimplementation of an existing tool, which has been extended with new features, a high level of customizability, a shorter processing time and an extensive evaluation on different text genres in English and Italian (i.e. scientific articles and historical texts).

Italiano. In questo articolo presentiamo un sistema di estrazione di espressioni-chiave chiamato Keyphrase Digger. Lo strumento usa sia misure statistiche che informazioni linguistiche per individuare una lista pesata di n-grammi corrispondenti ai concetti più importanti all'interno di un testo. KD è una reimplementazione di un sistema già esistente, che è stato esteso con nuove funzionalità, un alto livello di personalizzazione, maggiore velocità e una valutazione accurata su differenti generi testuali in inglese e italiano (i.e. articoli scientifici e testi storici).

1 Introduction

This paper presents *Keyphrase Digger* (henceforth KD), a new implementation of the KX system for keyphrase extraction. Both KX (Pianta and Tonelli, 2010; Tonelli et al., 2012) and KD combine statistical measures with linguistic information to identify and extract weighted keyphrases from English and Italian texts. KX took part to the SemEval 2010 task on “Automatic Keyphrase Extraction from Scientific Articles” (Kim et al.,

2010) achieving the 7th best result out of 20 in the final ranking. KX is part of TextPro (Pianta et al., 2008), a suite of NLP tools developed by Fondazione Bruno Kessler¹. The aim of KX re-implementation was to improve system performance in terms of F-measure, processing speed and customizability, so to make its integration possible in web-based applications. Besides, its adaptation to different types of texts has become possible also for not expert users, and its application to large document collections has been significantly improved.

Keyphrases are n-grams of different length, both single and multi-token expressions, which capture the main concepts of a given document (Turney, 2000). Their extraction is very useful when integrated in complex NLP tasks such as text categorization (Hulth and Megyesi, 2006), opinion mining (Berend, 2011) and summarization (D’Avanzo and Magnini, 2005)². Moreover keyphrases, especially if displayed using an effective visualization, can help summarize and navigate document collections so to allow their so-called ‘distant reading’ (Moretti, 2013). This need to easily grasp the concise content of a text through keyphrases is particularly relevant given the increasing availability of digital document collections in many domains. Nevertheless, outside the computational linguistics community, for example among humanities scholars, the extraction of keywords is often assimilated with the extraction of the most frequent (single) words in a text, see for instance the success of tools such as Textal³ and Voyant⁴ in the Digital Humanities community. In some cases, stopwords are still included among the top-ranked keywords, leading to a key-concept

¹<http://hlt-services2.fbk.eu/textpro/>

²For a comprehensive survey of the state of the art in automatic keyphrases extraction see Hasan and Ng (2014).

³<http://www.textal.org/how>

⁴<http://voyant-tools.org/>

list which is little informative and just reflects the Zipfian distribution of words in language. KD, instead, is designed to be easily customized by scholars from different communities, while sticking to the definition of keyphrases in use in the computational linguistics community.

The remainder of the paper is structured as follows. In Section 2 we describe KD architecture and features while in Section 3 the system evaluation is reported. We present the application of KX to the historical domain and the web-based interface available online in Section 4. Future works and conclusions are drawn in Section 5.

2 System Overview

KD is a rule-based system that combines statistical and linguistic knowledge given by PoS patterns. The system takes in input a text pre-processed with a tokenizer, a lemmatizer and a PoS tagger, and derives an ranked and weighted list of single words and multi-token expressions, which represent the most important concepts mentioned in a document. Differently from KX, whose pre-processing step was performed with TextPro for both Italian and English texts, English documents can now be pre-processed with Stanford CoreNLP (Manning et al., 2014), TreeTagger (Schmid, 1994) or TextPro, making KD more flexible than its ancestor.

Furthermore, KD is based on a parallel architecture implemented in Java. This constitutes a major efficiency improvement with respect to KX, which is implemented in Perl and has a sequential architecture. In particular, KX extracts first all possible n-grams from a text and applies selection rules only in a second phase, slowing down the extraction of appropriate candidates, especially in case of long documents. Instead, in KD the following five steps are performed:

1. A file is first split in n slices. The number of slices can be decided by the user or is automatically defined by the tool according to the number of machine's CPUs. Each part is processed by an isolated and parallel thread that extracts the set of n-grams corresponding to the language-dependent PoS patterns defined in a configuration file. This file contains the chains of meaningful PoS tags to be extracted, e.g. *noun+adjective* and *noun+preposition+noun* for Italian. Such sequences can be manually edited, deleted or

enriched by users, if necessary. The direct access to this configuration files (and also to the other configuration files) is realized by using the MapDB Java library⁵ that grants good performances at the read/write serialization time. Differently from KX, in this step the user can choose whether to run KD on inflected word forms or on lemmas, so to cluster extracted key-concepts (e.g. cluster *lingue straniere* and *lingua straniera* under the same key-concept).

2. A function merges the n-grams extracted from different threads in a common list. N-grams with a frequency lower than a threshold defined by the user are removed. In addition, frequencies are recalculated so that if a short key-concept is nested in a longer one (Frantzi et al., 2000) (e.g. *solidarietà economica* and *solidarietà economica internazionale*), the frequency of the former is deducted from the frequency of the latter.
3. The system checks whether, in the preliminary list of extracted concepts, some of them can be treated as synonyms. If yes, the corresponding entries are merged. Synonym resolution is performed on the basis of a list defined by the user, containing n-grams that the tool must consider equivalent, e.g. *liberismo* and *liberalismo economico*.
4. A first relevance score is computed for each concept in the list, taking into consideration different parameters that can be activated or disactivated by the user in a configuration file: frequency and inverse document frequency of n-grams, length of n-grams (so to prefer single words or multi-token expressions), position of first occurrence in the text, presence of specific suffixes (for example to give higher score to abstract concepts ending with *-ismo* and *-itudine*), boost of specific PoS patterns considered important in a given domain. This latter parameter is not present in KX. Another new feature is given by the integration of Apache Lucene library⁶: its scoring system allows to compute efficiently tf/idf at document level.

⁵<http://www.mapdb.org>

⁶<https://lucene.apache.org/core/>

5. If the user wants to give preference to specific (i.e. longer) key-concepts, a final re-ranking step can be included. In this way, key-concepts that are specific but have a low frequency are given more relevance than key-concepts that are generic and thus have a higher frequency. This re-ranking step was already present in KX, but the boosting effect had in our opinion an excessive impact on the final keyphrase list, possibly leading to the deletion of top-ranked unigrams. In KD the impact of the re-ranking has been limited to an adjustment of some weights.

3 Evaluation

The evaluation of KD covers different aspects of the system. First, we replicated the SemEval 2010 evaluation using the same data and scorer provided in the keyword extraction task. In this way we checked system performance in terms of F-measure, precision and recall on English texts and on a specific domain, namely scientific papers. As for Italian, we assessed the quality of keyphrases extracted from a corpus of historical documents against a set of key-concepts previously defined by an expert. In addition, we calculated the speed of KD to process a corpus of Italian texts.

In task 5 of SemEval 2010 evaluation campaign, systems were required to automatically assign keyphrases to a corpus of scientific articles and were assessed by using an exact match evaluation metric over stems. This means that micro-averaged precision, recall and F-score were calculated considering the top 5, 10 and 15 candidates found by participating systems that perfectly match the set of manually assigned gold standard keyphrases (in other words, no partial match was allowed). Given that criteria for keyphrase identification depend on the domain, KD parameters were configured to deal with scientific papers. We used the 144 training files and the corresponding answer keys to identify recurrent relevant PoS patterns not present in the default pattern list and determine which ones need to be boosted. On one side, we needed to give importance to long multi-token expressions (e.g. *unified utility maximization model*), which are typical of the scientific domain, on the other we needed to recognize and boost non-expanded acronyms (e.g. *cscw*) that play a central role in this type of articles. For this reason, a specific rule has been added to auto-

matically identify and give a higher weight to unigrams corresponding to acronyms. Furthermore, we noted that the majority of keyphrases provided as gold answers were bigrams and trigrams (74% of the total in the training), so we boosted their corresponding patterns. Overall, we found that the best system configuration on the training data was the following: *min frequency of occurrence* = 2; *max length of keyphrases* = 6; *IDF* = yes; *position of first occurrence at the beginning of the file* = yes; *use of Lucene scoring* = yes; *re-ranking algorithms* = no. Such configuration scored an F-measure of 27.5% on the training set (KX scored 25.6 on the same files). Table 3 shows the results obtained with the same configuration on the test set. Results over the 5, 10 and 15 top-ranked keyphrases are reported: the F-score for the top 15 candidates, i.e. 26.5%, corresponds to the second best results in SemEval 2010 competition with an improvement of almost 2 points with respect of KX performance (i.e. 23.9%). Note that the first-ranked system relied on a supervised approach, making KD the best performing rule-based system evaluated on this data set.

	Precision	Recall	F-score
Top 5	35.4%	12.7%	18.0%
Top 10	31.3%	21.4%	25.4%
Top 15	26.2%	26.8%	26.5%

Table 1: Precision, Recall and F-score of KD evaluated on the test set provided in task 5 of SemEval 2010

As for Italian, we asked a history scholar to manually identify a set of key-concepts considered relevant to characterize the corpus of Alcide De Gasperi’s writings, dating back to the first half of the XX century (De Gasperi, 2006)⁷. This task was performed independently from the development of KD, so no specific instructions related to the keyphrase extraction task were given (e.g. the scholar could select also keyphrases which were not present in the documents). A set of about 60 keyphrases was defined for each of the five relevant periods of De Gasperi’s political career, which we used as a gold standard to evaluate the system performance. Over these five periods, which correspond to five corpora, KD achieved a macro-average precision of 23.8% calculated in

⁷Alcide De Gasperi was the first Prime Minister of the Italian Republic and one of the founders of the European Union

an ‘exact match’ setting. Since some of the key-concepts identified by the expert do not appear in the text, it was impossible for KD to extract them. For instance, *Alleanza Atlantica* is an expression never used by De Gasperi who, instead, used the expression *Patto Atlantico*, correctly extracted by KD. We compared KD results with the ones obtained using the Distiller-CORE library developed by the University of Udine (De Nart et al., 2015) and available at <https://github.com/ailab-uniud/distiller-CORE>⁸. Distiller-CORE extracted 20 keyphrases from each of De Gasperi’s subcorpora, achieving a macro-average precision of 15%. Considering only the first 20 keyphrases extracted by KD against the full list of expert’s keywords, our tool achieved a precision of 42%.

Since KX speed was a main issue when processing large document collections, we also ran a comparison between KX and KD processing time, running both systems on the same corpus of 101,000 Italian tokens and on the same machine⁹. As for parameters, we used the most comparable setting: two re-rank algorithms, frequency of occurrence set at 1, max length of 4 tokens for extracted keyphrases. It took KD 7 seconds to return the list of keyphrases, whereas KX needed 3.4 minutes to complete the task¹⁰. The improved system speed makes KD particularly suitable for integration in web applications, where texts can be processed on the fly. Some examples are reported in the following section.

4 Applications

KD has been integrated in the last version of ALCIDE¹¹ (*Analysis of Language and Content In a Digital Environment*), an online platform for Historical Content Analysis (Moretti et al., 2014). In ALCIDE the output of KD is displayed by means of two visualizations: a bar chart and a tag cloud. This analysis and the corresponding visualizations are available both at the corpus and at the single document level. Moreover, the user can search for a specific key-concept, retrieve the documents where it appears and display its distribu-

tion along a timeline. Within ALCIDE, KD has been applied to a corpus of F.T. Marinetti’s writings (Daly, 2013), with the goal of exploring Futurism works with NLP tools. Figure 1 shows the 20 most frequent key-concepts extracted from all manifestos written by Marinetti between 1909 and 1921. Such key-concepts can be mainly divided into two categories: the ones related to the political program of Marinetti characterized by the exaltation of war and of his homeland (i.e. *guerra*, *Italia*, *patriottismo*, *eroismo*) and the ones associated with his artistic program, with particular emphasis on futurism style in poetry (*parole in libertà*) and theatre (*teatro della sorpresa*).

KD is also available as a web application at the link http://celct.fbk.eu:8080/KD_KeyDigger/, through which users can copy&paste sample documents and run the keyphrase extraction process. Four pre-defined parameter settings are available: one for scientific papers, one for historical texts, one for news articles and one for all the other types of texts. Besides, also single parameters can be further specified (e.g. maximum keyphrase length). Keyphrases can be visualized as bar chart and word cloud, and be exported in tab-separated format.

5 Conclusions

This paper presents KD, a keyphrase extraction system that re-implements the basic algorithm of KX but adds new features, a high level of customizability and an improved processing speed. KD currently works on English and Italian and can take in input texts pre-processed with different available PoS taggers and lemmatizers for these two languages. Nevertheless, the system could be easily adapted to manage more languages and additional PoS taggers by modifying few configuration parameters.

KD will be soon integrated in the next TextPro release¹² and it will be also released as a standalone module. Meanwhile, we made it available online as part of an easy-to-use web application, so that it can be easily accessed also by users without a technical background. This work targets in particular humanities scholars, who often do not know how to access state-of-the-art tools for keyphrase extraction.

⁸We also tried to use AlchemyAPI (<http://www.alchemyapi.com/api>) and Sensium (<https://www.sensium.io>) API endpoints but they do not allow processing long documents.

⁹CPU: 2.3GHz Intel Core i7, RAM: 8Gb 1600 mhz ddr3, Hard Disk: SSD serial SATA 3

¹⁰7,000 ms versus 206,546 ms

¹¹<https://youtu.be/PhkuOfIod1A>

¹²Check the TextPro website (<http://textpro.fbk.eu/> for updates

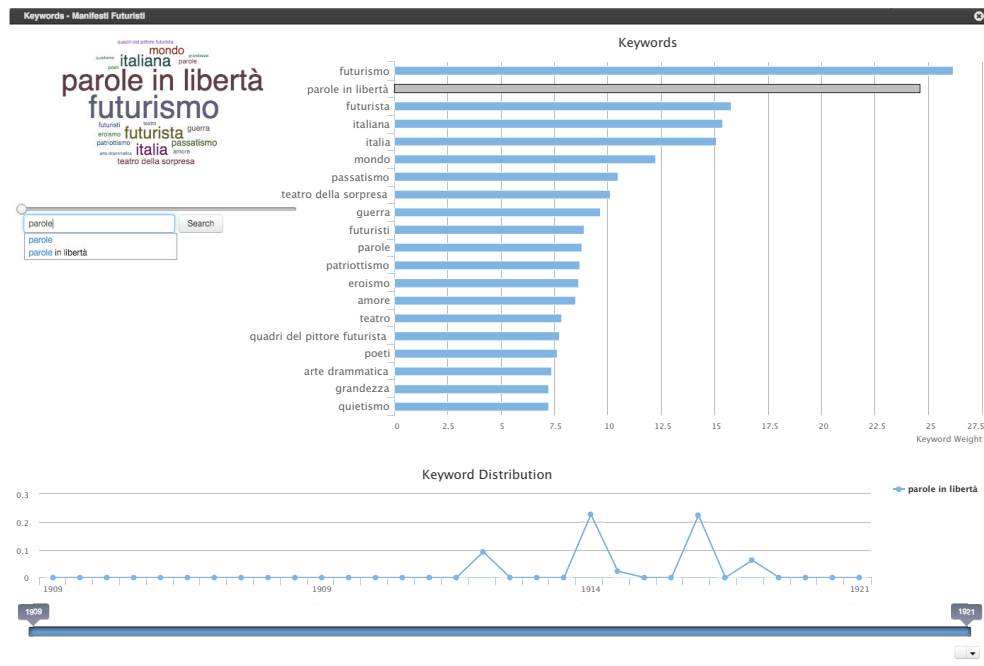


Figure 1: Visualization of key-concepts extracted with KD from Marinetti's manifestos in the ALCIDE platform

References

- Gábor Berend. 2011. Opinion Expression Mining by Exploiting Keyphrase Extraction. In *Proceedings of IJCNLP*, pages 1162–1170.
- Selena Daly. 2013. “The Futurist mountains”: Filippo Tommaso Marinetti's experiences of mountain combat in the First World War. *Modern Italy*, 18(4):323–338.
- Ernesto D'Avanzo and Bernado Magnini. 2005. A keyphrase-based approach to summarization: the lake system at DUC-2005. In *Proceedings of DUC*.
- A. De Gasperi. 2006. Scritti e discorsi politici. In E. Tonezzer, M. Bigaran, and M. Guiotto, editors, *Scritti e discorsi politici*, volume 1. Il Mulino.
- Dario De Nart, Dante Degl'Innocenti, and Carlo Tasso. 2015. Introducing distiller: a lightweight framework for knowledge extraction and filtering. In *Proceedings of the UMAP Workshops*.
- Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. 2000. Automatic recognition of multi-word terms: the C-value/NC-value method. *International Journal on Digital Libraries*, 3(2):115–130.
- Kazi Saidul Hasan and Vincent Ng. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the Association for Computational Linguistics (ACL)*.
- Anette Hulth and Beáta B Megyesi. 2006. A study on automatically extracted keywords in text categorization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 537–544.
- Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26.
- Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.
- Giovanni Moretti, Sara Tonelli, Stefano Menini, and Rachele Sprugnoli. 2014. ALCIDE: An online platform for the Analysis of Language and Content In a Digital Environment. In *Atti della prima Conferenza Italiana di Linguistica Computazionale*.
- Franco Moretti. 2013. *Distant Reading*. Verso, London.
- Emanuele Pianta and Sara Tonelli. 2010. KX: A flexible system for keyphrase extraction. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 170–173.
- Emanuele Pianta, Christian Girardi, and Roberto Zanolli. 2008. The TextPro Tool Suite. In *Proceedings of the Language Resources and Evaluation Conference*.

- Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the international conference on new methods in language processing*, volume 12, pages 44–49.
- Sara Tonelli, Elena Cabrio, and Emanuele Pianta. 2012. Key-concept extraction from french articles with KX. *Actes de latelier de clôture du huitième défi fouille de texte (DEFT)*, pages 19–28.
- Peter Turney. 2000. Learning Algorithms for Keyphrase Extraction. *Information Retrieval*, 2(4):303–336.