Extracting Keyphrases from Web Pages

Felice Ferrara and Carlo Tasso

Artificial Intelligence Lab.,
Department of Mathematics and Computer Science
University of Udine, Italy
{felice.ferrara,carlo.tasso}@uniud.it

Abstract. Social tagging systems allow people to classify Web resources by using a set of freely chosen terms commonly called tags. However, by shifting the classification task from a set of experts to a larger and untrained set of people, the results of the classification are not accurate. The lack of control and guidelines generates noisy tags (i.e. tags without a clear semantic) which lower the precision of the user generated classifications. In order to face this limitation several tools have been proposed in the literature for suggesting to the users tags which properly describe a given resource. On the other hand we propose to suggest n-grams (named keyphrases) by following the idea that sequences of two/three terms can better face potential ambiguities. More specifically, in this work, we identify a set of features which characterize n-grams adequate for describing meaningful aspects reported in the Web pages. By means of these features, we developed a mechanism which can support people when classifying Web pages by automatically suggesting meaningful keyphrases.

1 Introduction

Jeff Howe defined social tagging systems as one of the main examples of *crowdsourcing* systems [8]. Coined by Howe in June 2006, the term crowdsourcing appeared the first time in the article '*The Rise of Crowdsourcing*' [1] for defining the act of sourcing tasks traditionally performed by specific individuals (with specific competences) to an undefined large community of people (the crowd). According to Howe's theory the technological advances can significantly reduce the gap between professionals and amateurs: people can use cheap technologies to execute complex tasks. In this way complex tasks, such as the classification of digital resources, can be executed by a large community of people by saving significant resources: this is clearly achieved at the cost of less accurate results. Money is not the only way to compensate the crowd for their work: prizes, services or the intellectual satisfaction can stimulate people to use their intelligence and talent into sophisticated tasks.

The large population of the users of social tagging systems are the crowd used to classify Web resources on behalf of knowledge engineers and domain experts. Social tagging systems do not provide a monetary compensation to the taggers, but people are compensated with:

M. Agosti et al. (Eds.): IRCDL 2012, CCIS 354, pp. 93–104, 2013.

- The services provided by the social tagging system. Social collaboration is a good mean for retrieving meaningful information since each user can enjoy the classification produced by other peers. Moreover, by tagging resources, people can easily retrieve resources they classified in the past.
- Intellectual satisfaction. Users can be interested in using social systems to propagate their ideas, to influence other people, and to help people with similar information needs.

Obviously, by shifting the classification task from a set of experts to a larger set of untrained people, the results of the classification cannot be rigorous. In fact, due to the lack of control and guidelines, the precision of the returned classification produced is lowered by noisy tags (i.e. tags without a clear semantic).

How can we reduce the gap between experts and Web users? The answer to this question is still in Howe's ideas of filling the gap between people with specific expertise and not experts with proper technologies: according to this theory we can reduce the gap between knowledge engineers and users of social tagging systems by introducing tools able to simplify the classification task.

In order to reach this aim, we can support people with mechanisms able to suggest significant and appropriate tags which can be used to classify Web resources in a adequate way. In this work we propose to suggest to the user multi-terms, i.e. n-grams named keyphrases, as a support for classification. The main motivation to suggest keyphrases is that many concepts are reported as multi-terms (for instance the concept '*Unified Modeling Language*'). In these cases, keywords (i.e. uni-grams) do not properly represent the concepts which should be used to label/classify digital document. Following this idea, we propose in this paper the DIKpEW (Domain Independent Keyphrase Extraction for Web pages) mechanism which is aimed at supporting people classifying Web pages by extracting potentially relevant and significant n-grams from the content of the specific considered HTML page. Obviously the proposed system cannot substitute the work of experts, but it is a tool usefull to normalize the user classifications by reducing the number of ambiguous/misleading classifications.

The paper is organized as follows: in Section 1.1 we survey the keyphrase extraction task; the proposed approach to extract keyphrases from Web pages is illustrated in Section 2; Section 3 describes the evaluation settings and the results; final considerations conclude the paper in Section 4.

1.1 Keyphrases Extraction

Keyphrase extraction methods have been successfully used for executing relevant tasks in the field of digital libraries, such as: indexing document collections [7], classifying resources [14], providing automatic tagging [19], and filtering resources [6,16]. The task of extracting keyphrases from textual resources is usually implemented in two steps: the *candidate identification* phase and the *selection* phase. The candidate identification phase is exploited in order to identify an initial set of possible keyphrases for a given document. This initial set of keyphrases (referred as '*candidate keyphrases*') is then analyzed in the selection phase for

selecting only the most meaningful ones, i.e. the candidates keyphrases which better summarize the textual resource. Existing methods for keyphrase extraction can be divided into supervised and unsupervised approaches.

A *supervised approach* builds a model by using training documents that have already keyphrases assigned by humans. This model is trained to learn features of the relevant keyphrases (the keyphrases assigned by humans to the training documents) and then it is exploited in order to select keyphrases from previously unseen documents. *KEA* [22] is a notable supervised approach which uses a Bayesian classifier. KEA analyzes training documents by taking into account orthographic boundaries (such as punctuation marks and newlines) in order to find candidate phrases. In KEA two specific features are exploited: tf×idf (term frequency × inverse document frequency) and the position of the first occurrence of the term. Hulth [9] introduces linguistic knowledge (i.e., *POS, Part-Of-Speech tags*) in determining candidate sets: 56 potential *pos-patterns* are used for identifying candidate phrases in the text. The experimentation carried out by Hulth has shown that, using a *POS tag* as a feature in candidate selection, a significant improvement of the keyphrase extraction results can be achieved. Another system that relies on linguistic features is LAKE (Learning Algorithm for Keyphrase Extraction) [5]: it exploits linguistic knowledge for candidate identification and it applies a Naive Bayes classifier in the final keyphrase selection. All the above systems need training data (in a larger or smaller extent) in order to construct an extraction system. However, acquiring training data with known keyphrases is not always feasible and human assignment is time-consuming. Furthermore, a model that is trained on a specific domain, does not always produce adequate classification results in other domains.

The *unsupervised approach* eliminates the need of training data. It selects a general set of candidate phrases from the given document, and it uses some ranking strategy to select the most important candidates as keyphrases for the document. Barker and Cornacchia [2] extract noun phrases from a document and ranks them by using simple heuristics, based on their length, frequency, and the frequency of their head noun. In [3], Bracewell et al. extract noun phrases from a document, and then cluster the terms which share the same noun term. The clusters are ranked based on term and noun phrase frequencies. Finally, the top-n ranked clusters are selected as keyphrases for the document. The authors of [17] and [15] proposed unsupervised approaches based on a graph representation of documents. Such approaches use ranking strategies (similar to the PageRank algorithm [4]) to assign scores to each term. Keyphrase extraction systems that are developed by following unsupervised approaches are in general domain independent since they are not constrained by specific training documents.

2 Extracting Keyphrases from Web Pages

In [20] we proposed an approach for extracting keyphrases from scientific papers showing also that it outperforms other state of the art mechanisms. The approach we proposed in [20] works under two main assumptions:

1. **A large part of scientific papers is usually written in English**. This simplifies the analysis of the textual content since we have to take into account only the characteristics of the English language.
2. **Scientific papers organize their contributions according to a well-defined schema**. The abstract, the introduction and the conclusion are the sections where the authors usually summarize the goals, the issues and the findings of the work. For this reason, we assign a score to each keyphrase by evaluating the position of the keyphrase in the text: it is plausible that keyphrases in the first part and in the last section of the paper better describe the resource.

These two assumptions are not always true when we want to extract keyphrases from Web pages. In fact, Web pages can be written in languages different from English and, moreover, Web pages do not follow the structure normally adopted by scientific papers. The main aim of this work is to extend, to modify, and to improve the approach we proposed in [20] in order to extract keyphrases from Web pages.

The workflow of DIKpEW, the mechanism proposed in this paper, is shown in Figure 1. By following the traditional schema adopted by keyphrase extraction mechanisms we split the workflow in two parts focused respectively on candidate phrase extraction and on phrase selection phase, described in the following two subsections.

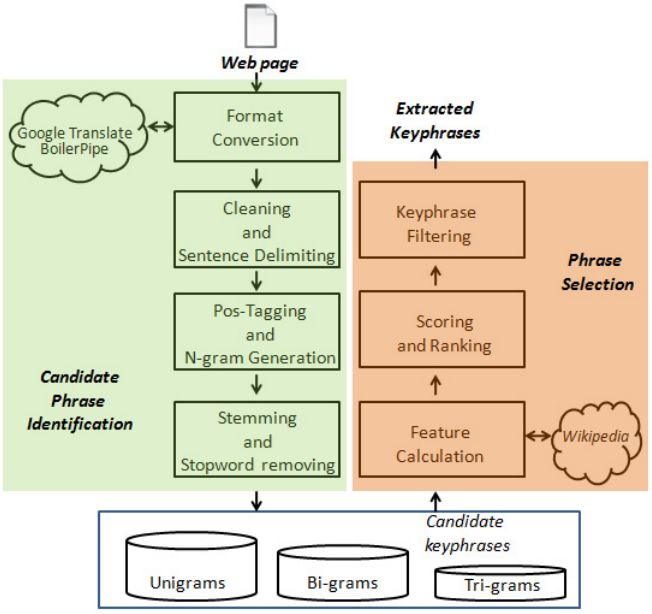

Fig. 1. The workflow used for extracting keyphrases from Web pages

2.1 DIKpEW: Candidate Phrase Identification

Given an HTML page, a format conversion step is exploited for extracting the meaningful textual corpus from the document, i.e the textual parts which contain the relevant facts reported in the resource. More specifically, the format conversion is aimed at:

- removing the irrelevant parts from the document. Unfortunately, the main contents of Web pages are often mixed with other textual parts (typically in the headers, the footers, etc.) which are completely irrelevant. In order to discard these useless and noisy parts from the Web page we use an open source Web service called Boilerpipe[1]. The Boilerpipe service, developed by researchers from the L3S Research Center of Hannover, can remove the '*surplus*' text from a Web page. Given a Web page, Boilerpipe returns the main text in the Web page by discarding other information (banner, footers, advertisement, etc.).
- Extracting metadata included by the authors of the Web page. HTML pages are often enriched by their authors with some labels and summaries. These metadata are stored by using tags of the HTML language (*KEYWORDS*, *DESCRIPTION*, and *TITLE* tags).
- Translating the text into the English language. We cannot assume that Web pages are always written into English. In order to re-use the POS-Tagger as well as the POS-Patterns adopted in [20], we translate the text extracted by the Boilerpipe service into English. Currently, we use the Google Translate Api in order to recognize the input language and to translate the text in English.

The output of the format conversion phase is a text in English constituted by the title of the Web page, followed by the metadata extracted from the HTML tags, and concluded by the text extracted by the Boilerpipe service.

This text is analyzed in the cleaning and sentence delimiting step in order to delimit sentences, following the assumption that a keyphrase cannot be located simultaneously in two distinct sentences.

In the POS-tagging and n-gram extraction step we assign a POS tag (noun, adjective, verb, etc.) to each token in the cleaned text by using the Stanford log-linear part-of-speech tagger[2] and then we extract all possible subsequences of phrases including up to 3 words (uni-grams, bi-grams, and tri-grams).

A pruning process is exploited in the stemming and stopword removing step in order to discard keyphrases which do not have a very significant meaning. To this aim, we remove the phrases that start and/or end with a stopword and the phrases containing a sentence delimiter. Partial stemming (i.e., unifying the plural forms and singular forms which refer essentially to the same concept) is performed using the first step of Porter stemmer algorithm [18]. We do not exploit the other steps of the Porter stemmer since they are not appropriate for

[1] `http://code.google.com/p/boilerpipe/`
[2] `http://nlp.stanford.edu/software/tagger.shtml`

keyphrase extraction (consider, for example, the removal of the '*ing*' suffix in the bi-gram '*software engineering*'). To further reduce the size of the candidate phrase set, we filter out some candidate phrases by using POS tagging information: Uni-grams that are not labeled as noun, adjective, or verb are filtered out. For bi-grams and tri-grams, only the POS-patterns defined by Justeson and Katz [13] and other patterns that include adjective and verb forms are considered.

Generally, in a document, uni-grams are more frequent than bi-grams, and bi-grams are more frequent than tri-grams, and so on. For taking into account this phenomenon, we build three lists, containing uni-grams, bi-grams, or tri-grams respectively. This allows to treat them separately, without any bias towards uni-grams with respect to bi-grams and tri-grams.

2.2 DIKpEW: Phrase Selection

As in [20], some characteristics of the candidate keyphrases are assessed in the feature calculation step for identifying the most relevant keyphrases. The evaluated characteristics have been identified by taking into account how usually Web pages store meaningful information. The considered features are listed and described in following.

1. ***Phrase frequency***: this feature is the classical term frequency (TF) metric, exploited in many state of the art keyphrase extraction systems [21][9][10]. In our work, the TF value is normalized with respect to the specific n-gram list. More specifically, given the phrase P in the list L (the list of unigrams, bi-grams or tri-grams) we define

$$frequency(P, L) = \frac{freq(P, L)}{size(L)}$$

   where $freq(P, L)$ is the number of times P occurs in L and $size(L)$ is the total number of phrases included in L.
2. ***POS value***: as observed in [9] and [2], most author-assigned keyphrases for a document turn out to be noun phrases. For this reason, in our approach, we stress the presence of nouns in candidate phrases by computing POS value as the ratio of the number of nouns in the keyphrase by the total number of terms in the keyphrase.
3. ***Phrase depth***: this feature reflects the belief that very frequently Web pages report the most relevant facts at the very beginning of the document: some statistics identify the initial 25% of the text as the part where all main concepts and information are usually reported [12]. In order to highlight such phrases we compute the phrase depth value for phrase P in a document D as:

$$depth(P, D) = 1 - \left[\frac{first\_index(P)}{size(D)}\right]$$

   where $first\_index(P)$ is the number of words preceding the phrase's first occurrence and $size(D)$ is the total number of words in D. The result is

a value in $[0, 1]$ and highest values are assigned to phrases reported in the initial part of the document.

4. ***Wikipedia***. The Wikipedia feature is used to identify more coherent and recognized phrases by following the idea that keyphrases associated to articles in the Wikipedia encyclopedia are more likely associated to well-defined concepts/meaning. The Wikipedia feature is then set to 1 if Wikipedia has a page for describing the keyphrase, 0 otherwise.
5. ***Title***. It highlights keyphrases that are included in the title of the Web page (if known). We followed the hypothesis that the title summarizes meaningful concepts which are more deeply discussed in the rest of the text. For each keyphrase, we compute a boolean feature which is set to 1 if the keyphrase is in the title of the Web page, 0 otherwise.
6. ***Description***. Authors of Web pages often add a short description of the main contents of the Web page by using the *DESCRIPTION* HTML tag. According to the idea that the summary provided by the author may contain very meaningful information we compute this boolean feature for each keyphrase: the feature is set to 1 if the keyphrase is in the description, 0 otherwise.
7. ***Keyword***. Even if authors of Web pages are not required to classify their published resources, they usually add some keywords in order to be properly indexed by search engines. Since these terms are labels generated by the authors themself, we consider these terms as meaningful keyphrases. The keyword feature is then computed as a boolean value which is set to 1 if the keyphrase is one of the keywords proposed by the author of the Web page, 0 otherwise.

In the scoring and ranking step, all the above features are used in order to compute a score (named *keyphraseness*) for each candidate keyphrase. The keyphraseness is a weighted combination of the evaluated features, and in particular, given a candidate keyphrase p, the keyphraseness is computed as

$$keyphraseness(p) = \sum_i w_i * f_i(p)$$

where: $f_i(p)$ is the value of the i-th feature for p and w_i is the weight assigned to the i-th feature.

A preliminary experimentation was carried out for identifying a proper set of weights for the features: a first prototype was implemented for collecting the opinions of a restricted set of subjects about the accuracy of the extracted keyphrases. By using this feedback, we identified the weights currently assigned to the features, which are the same for uni-grams, bi-grams, and tri-grams. However, future work will also investigate the idea of using different weights for uni-grams, bi-grams, and tri-grams since they have different characteristics. For example, unigrams extracted from a Web page are more frequent than bi-grams and trigrams. This preliminary experimentation allowed us to identify the the weights of the features reported in Table 1.

Table 1. The weights assigned to the features

Feature Name	Weight
phrase frequency	0.5
POS value	0.5
phrase depth	0.6
wikipedia	0.9
title	0.9
description	0.6
keyword	0.6

The weights shown in Table 1 are used to compute the keyphraseness of the candidate phrases extracted from Web pages and then, the obtained lists of unigrams, bi-grams, and tri-grams, are ranked according to their keyphraseness.

Finally, the keyphrases associated with higher scores (higher keyphraseness) are recommended in the final keyphrase filtering step. We decided to extract the two top scored unigrams, the five top scored bi-grams, and the three top scored tri-grams since this setting generated the best results during a preliminary analysis. The reader can also notice that we use keyphraseness only for ordering the keyphrases and for this reason we do not need to normalize the keyphraseness in $[0, 1]$.

3 Evaluation

Web pages are usually not classified with keyphrases by their authors and this lack had a strong impact on our evaluation procedure. In fact there are not freely available datasets which can be used to execute an automatic evaluation of the described mechanism. For this reason we decided to exploit a live evaluation involving a set of volunteers which had the task of judging the accuracy of the results returned by our approach. Moreover, due to the lack of keyphrases associated to Web pages, we could not use KEA for comparing our results to one of the state of the art mechanisms. In fact, the KEA mechanism needs to be trained by using a corpus of annotated documents. This is a strong limitation since, at the best of our knowledge, there are not freely available APIs for extracting ranked keyphrases from Web pages. In order to face this issue we decided to use as baseline approach a system where keyphrases are scored and ranked according only to their frequencies. This choice seems reasonable since, as our approach does, the baseline approach takes into account only the information available in a specific document (without considering the characteristics of the documents in a specific collection). This baseline mechanism is still domain independent and the results are not biased by the characteristics of a specific corpus. More specifically, the baseline mechanism assigns a score to the set of candidate keyphrases according to their frequency: the most frequent keyphrases obtain an higher score. By using the score assigned to keyphrases, the baseline mechanism can extract the two top scored uni-grams, the five top

scored bi-grams, and the three top scored tri-grams. The final set of keyphrases is then built by these 10 filtered keyphrases.

The results returned by both our mechanism and the baseline approach were evaluated by using a Web application where a set of volunteers judged the accuracy of the results. Since our approach is mainly aimed at supporting the users of social tagging systems, we built a Web based application which simulates the interaction of a user with a social tagging system. By using this application, the volunteers could submit an URL and then the evaluation framework returned to the users a list of suggested keyphrases for the specific Web page. The list of returned keyphrases was built by merging the results produced by both the proposed approach and the baseline mechanism. However, the two sets of keyphrases were presented to the evaluators in a random order.

By merging the keyphrases without a specific order we avoided to bias the human evaluators since they were not able to recognize the keyphrases returned by one of the two compared approaches.

The evaluators had to vote each returned keyphrase by using the following 5-Likert scale: **Excellent** - The keyphrase is very meaningful, it reports relevant facts, people, topics or other elements which characterize the Web page; **Good** - The keyphrase is still significant for classifying the document, but it is not the best: the keyphrase reports facts, people, topics or other elements which characterize the Web page, but are more weakly connected to the main content of the page; **Neutral** - You are not sure about the significance of the keyphrase for the document; **Poor** - The keyphrase does not properly describe the contents; **Very Poor** - The keyphrase does not make sense.

The evaluation involved 26 volunteers (20 men and 6 women) who worked for two weeks. The volunteers were students and workers. The oldest participant was 63 years old, the youngest was 22 years old and the average age was 37 years. The volunteers evaluated the keyphrases generated for 209 Web pages written in Italian and in English.

We used the Normalized Discounted Cumulative Gain (NDCG) metric [11] to evaluate the experimental results. The NDCG metric is commonly used in Information Retrieval in order to evaluate the accuracy of ranking mechanisms. This measure is specifically used in scenarios where the ranked results are associated to different relevance levels, since it takes into account both the position and the usefulness (or gain) of the results. In other words, the NDCG metric evaluates a raking mechanism according to its capability of placing the most relevant resources in the higher positions of the generated ranking. Technically, given a ranked list of resources returned by the evaluated mechanism, where the resource (in our case the keyphrase) in position i is associated to a relevance level rel_i (in our case the position is defined by our algorithm and the relevance by one of the evaluators), the NDCG computes the gain for this list as follows

$$DCG = rel_1 + \sum_{i=2}^{n} \frac{rel_i}{log_2 i}$$

where n is the number of results in the ranked list and in our specific case n is equal to 10. In our evaluation the graded relevance scale is defined by the following relevance levels: Excellent = 4; Good = 3; Neutral = 2; Poor = 1; Very poor = 0. The DCG is then used to quantify the accuracy of a response generated by a ranking mechanism according to both a fixed relevance scale and the opinions of an evaluator.

By computing the DCG over each evaluation provided by our evaluators, we obtained an assessment of the accuracy for each evaluated Web page. These DCGs are then normalized with respect to the ideal rankings (i.e., the DCGs of the rankings generated by placing the most relevant results in the higher positions)to compute the NDCG and a higher NDCG corresponds to a more accurate approach.

Table 2 reports the 8 different NDCG values computed for evaluating and comparing the accuracy of the top 5 and top 10 keyphrases extracted by: (i) our approach from Web pages written in Italian (DIKpEW_Ita); (ii) the baseline system from Web pages written in Italian (Base_Ita); (iii) our approach from Web pages written in English(DIKpEW_Eng); (iv) the baseline system from Web pages written in English(Base_Eng).

Table 2. Performance of DIKpEW compared to the baseline mechanism

	NDCG@5	**NDCG@10**
Base_Ita	0.484	0.437
DIKpEW_Ita	0.558	0.614
Base_Eng	0.485	0.576
DIKpEW_Eng	0.523	0.686

According to the results showed in the table our approach outperforms the baseline mechanism. Moreover, the accuracy of the results computed for the Web pages in Italian are comparable to the accuracy for the Web pages in English. This means that the noise introduced by the translation in English does not significantly lowers the accuracy of the results. This can be justified in two ways: (i) the weight of the keyphrase depends on a set of statistical features which discard possible incorrect translations; (ii) the Wikipedia feature allows us to throw out (or at least to assign to lower positions) the bi-grams and tri-grams which have not a clear meaning.

4 Conclusion

In this work we presented an approach which is aimed at supporting the users of social tagging systems in classifying Web pages. In particular, the proposed approach identifies n-grams from a Web document for suggesting meaningful labels for the specific resource. An experimental evaluation showed that the proposed approach is plausible and future analysis will investigate if the proposed

approach can produce better results for specific topics or specific sets of Web pages (blogs, newspapers, etc.).

The proposed approach can extract keyphrases which appear already in a given document. Future work will focus on overcoming this limitation by navigating other knowledge sources such as Wikipedia, Wordnet or a specific domain ontology. In such a way it is possible to produce meaningful tags constituted by uni-grams, bi-grams, and tri-grams which are not contained in the text, and that are the result of a domain reasoning activity. A future work will investigate the problem of identifying a suitable threshold in the value of keyphraseness above/below which to accept/reject a candidate keyphrase.

References

1. The rise of crowdsourcing. Website, `http://www.wired.com/wired/archive/14.06/crowds.html`
2. Barker, K., Cornacchia, N.: Using Noun Phrase Heads to Extract Document Keyphrases. In: Hamilton, H.J. (ed.) Canadian AI 2000. LNCS (LNAI), vol. 1822, pp. 40–52. Springer, Heidelberg (2000)
3. Bracewell, D.B., Ren, F., Kuroiwa, S.: Multilingual single document keyword extraction for information retrieval. In: Proceedings of the 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering, Wuhan, pp. 517–522 (2005)
4. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks 30(1-7), 107–117 (1998)
5. D'Avanzo, E., Magnini, B., Vallin, A.: Keyphrase extraction for summarization purposes: the lake system at duc2004. In: DUC Workshop, Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics Annual Meeting, Boston, USA (2004)
6. Ferrara, F., Pudota, N., Tasso, C.: A Keyphrase-Based Paper Recommender System. In: Agosti, M., Esposito, F., Meghini, C., Orio, N. (eds.) IRCDL 2011. CCIS, vol. 249, pp. 14–25. Springer, Heidelberg (2011)
7. Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-Manning, C.G.: Domain-specific keyphrase extraction. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, pp. 668–673. Morgan Kaufmann Publishers, San Francisco (1999)
8. Howe, J.: Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business, 1st edn. Crown Publishing Group, New York (2008)
9. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, pp. 216–223. Association for Computational Linguistics, Morristown (2003)
10. Hulth, A., Megyesi, B.B.: A study on automatically extracted keywords in text categorization. In: ACL-44: Proc. of the 21st Int. Conf. on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, vol. 44, pp. 537–544. ACL, Morristown (2006)
11. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Transaction on Information Systems 20(4), 422–446 (2002)
12. Jones, S., Paynter, G.W.: An evaluation of document keyphrase sets. Journal of Digital Information 4(1) (2003)

13. Justeson, J., Katz, S.: Technical terminology: some linguistic properties and an algorithm for identification in text. Natural Language Engineering 1, 9–27 (1995)
14. Krulwich, B., Burkey, C.: Learning user information interests through the extraction of semantically significant phrases. In: Hearst, M., Hirsh, H. (eds.) AAAI 1996 Spring Symposium on Machine Learning in Information Access, pp. 110–112. AAAI Press, California (1996)
15. Litvak, M., Last, M.: Graph-based keyword extraction for single-document summarization. In: Proceedings of the Workshop on Multi-Source Multilingual Information Extraction and Summarization, pp. 17–24. ACL, Morristown (2008)
16. Micarelli, A., Gasparetti, F., Biancalana, C.: Intelligent Search on the Internet. In: Stock, O., Schaerf, M. (eds.) Reasoning, Action and Interaction in AI Theories and Systems. LNCS (LNAI), vol. 4155, pp. 247–264. Springer, Heidelberg (2006)
17. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: Dekang, L., Dekai, W. (eds.) Proc. of Empirical Methods in Natural Language Processing, pp. 404–411. Association for Computational Linguistics, Barcelona (2004)
18. Porter, M.F.: An algorithm for suffix stripping. In: Readings in Information Retrieval, pp. 313–316 (1997)
19. Pudota, N., Dattolo, A., Baruzzo, A., Ferrara, F., Tasso, C.: Automatic keyphrase extraction and ontology mining for content-based tag recommendation. International Journal of Intelligent Systems, Special Issue: New Trends for Ontology-Based Knowledge Discovery 25, 1158–1186 (2010)
20. Pudota, N., Dattolo, A., Baruzzo, A., Tasso, C.: A New Domain Independent Keyphrase Extraction System. In: Agosti, M., Esposito, F., Thanos, C. (eds.) IRCDL 2010. CCIS, vol. 91, pp. 67–78. Springer, Heidelberg (2010)
21. Turney, P.: Learning to extract keyphrases from text. Technical Report ERB-1057, National Research Council, Institute for Information Technology (1999)
22. 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, New York (1999)