Towards Building a Standard Dataset for Arabic Keyphrase Extraction Evaluation

Muhammad Helmy*, Marco Basaldella*, Eddy Maddalena*, Stefano Mizzaro* and Gianluca Demartini†
*University of Udine, Udine, Italy
Email: {alameldien.muhammad, basaldella.marco.1}@spes.uniud.it, {eddy.maddalena, mizzaro}@uniud.it
†University of Sheffield, Sheffield, UK
Email: g.demartini@sheffield.ac.uk

***Abstract*—Keyphrases are short phrases that best represent a document content. They can be useful in a variety of applications, including document summarization and retrieval models. In this paper, we introduce the first dataset of keyphrases for an Arabic document collection, obtained by means of crowdsourcing. We experimentally evaluate different crowdsourced answer aggregation strategies and validate their performances against expert annotations to evaluate the quality of our dataset. We report about our experimental results, the dataset features, some lessons learned, and ideas for future work.**

***Keywords*-Arabic Language Resources; Dataset; Keyphrase Extraction; Crowdsourcing;**

I. Introduction

While the problem of automatically indexing documents using keywords has been studied for more than 50 years [1], the task of Automatic Keyphrase Extraction (henceforth KPE) came to the attention of the research community in the late 1990s. [2] identified several purposes for keyphrases (henceforth KPs), which included summarization, to offer a quick glimpse of a document's content, indexing, when used, e.g., in the index of a journal, and search engine aid, when used to label documents for information retrieval purposes.

Since then, many algorithms for KPE have been developed, which can be put in two categories: *supervised* and *unsupervised* approaches [2], [3], [4]. The supervised approach requires a training dataset for its machine learning algorithm and both of them require a gold standard to evaluate the extracted KPs. Many datasets have been proposed in the past years, prominently for the English language. The SEMEVAL 2010 dataset [5] is widely used for evaluating the performance of KPE algorithms, along with other datasets as those proposed by [6] or [7].

To our knowledge, all the well-formed datasets available cover only the English language. This fact obviously hinders the development of a multi-lingual KPE community. For example, there is a growing interest around the problem of KPE in the Arabic language. Arabic is, in fact, the fifth most spoken language in the world, with more than 240 million native speakers[1]. Nevertheless, there is no shared, standard dataset that scholars can use to assess the performance of their algorithms. For example, [4] and [8] used custom-made corpora in their work, remarking the absence of a standard dataset.

The aim of this work is then to provide a dataset for the growing community of Arabic KPE to train and evaluate their KPE algorithms.

[1] https://www.ethnologue.com/statistics/size

II. Related Work

[4] performed an evaluation over manually annotated documents including Wikipedia articles and their meta-tags to train and evaluate KP-Miner. [8] used a mixed approach of previously tagged document and manually annotated ones as a dataset. However, in this dataset, only 73% of the human-generated keywords are actually found in the text, severely undermining the quality of the dataset and the performance of the KPE algorithm itself.

The use of crowdsourcing in the KPE field is very recent. [3] collected a KPE dataset for English using Amazon Mechanical Turk. [9] used crowd workers to rate the quality of automatically selected KPs. Employing crowdsourcing for Arabic natural language processing tasks produced mixed results so far. [10] showed that the quality of Arabic workers available in Mechanical Turk was not satisfying enough for POS (Part Of Speech) tagging or grammatical case annotation. However, [11] used with satisfaction the same platform for building an Arabic corpus for the much easier task of text summarization.

In our work we use the crowdsourcing platform Crowdflower to generate an Arabic KP collection with the support of 226 workers.

III. The Crowdsourcing task

A. Document Collection

The document collection we used contains 160 documents selected from four general purpose, freely available corpora: 46 documents from Arabic Newspapers Corpus (ANC) [12], 53 from Corpus of Contemporary Arabic (CCA) [13], 31 from Essex Arabic Summaries Corpus (EASC) [14], [11], and 30 from Open Source Arabic Corpora (OSAC) [15].

The documents are categorized into nine topics: art and music, environment, finance, health and medicine,

978-1-5090-0922-0/16/$31.00

Table I: Examples of Arabic words and their lemmatized forms.

Word	Meaning	Lemma	Meaning
مدرستنا	Our School	مدرسة	A School
المدرسون	The Teachers	مدرس	A Teacher
الدارسون	The Students	دارس	A Student
دروس	Lessons	درس	A Lesson
دراستهم	Their Study	دراسة	A Study

politics, religion, science and technology, sport, and tourism. Each category includes from 17 to 19 documents. After preprocessing the documents, by eliminating unrelated text like headers, image captions, and corpus metadata, their lengths vary between 500 and 1000 words, with a median of 735.5 words.

The analysis process requires three different forms of Arabic text for both documents and selected KPs. The first one is the original form, which is the text without processing or removing any character. The second one, called "pure form", includes only Arabic alphabet and numbers. In other words, "pure" KPs are the selected phrases with Arabic diacritic signs and non-Arabic characters removed.

For a more in-depth analysis, we extracted the root form of the words, i.e. we removed any information about gender, number, pronouns, etc. attached to the word. Usually, for the English language, a stemming algorithm is used to perform this task. In Arabic, however, stemming is much less effective than in other languages, since it simply removes the derivational affixes of the word and is often not able to get its true root (stem). Thus, many words with different meanings would be reduced to the same stem.

To tackle this problem, we decided to use lemmatization and not stemming. Lemmatization, performed with the AraMorph[2] tool, applies vocabulary and morphological analysis on the word to get its dictionary form (lemma), resulting in a much more precise "cut" of the originally selected words. This way we could obtain a standard root form of each selection, which allowed us ultimately to merge similar selections together. Table I provides an example of such technique for a set of Arabic words. All of these words have the same Arabic stem which is (درس, translated as "study"), but have different lemmas.

B. Keyphrase Collection

First of all, we launched a pilot experiment on the Crowdflower platform with 10 documents, to tune our task for the whole corpus. The results of the pilot experiment helped us to tune the experimental design for the actual Corpus Collection. We decided to use 10 workers per document, and to ask each worker to select 10 KPs, while in the pilot experiment we required just 5 KPs by 5 workers. Moreover, we adjusted the task instructions to guide workers to not select phrases beginning with stopwords, verbs, or adjectives. Finally, each unit was discarded if the worker did not spent at least 120 seconds on the document. Each worker could read and annotate up to ten documents, i.e., could complete up to ten units. Finally, we required Crowdflower to select only medium and highest quality workers.

[2]http://www.nongnu.org/aramorph

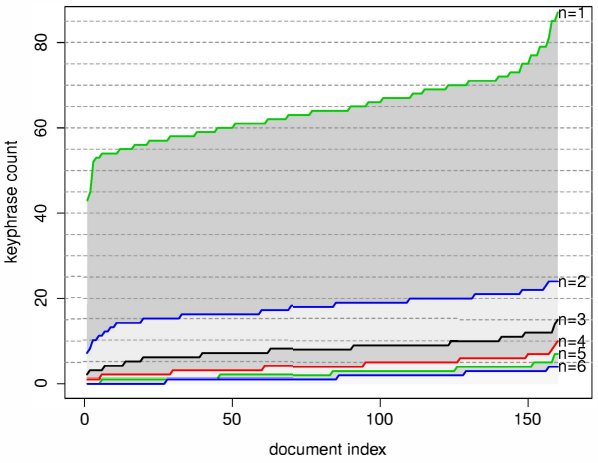

Figure 1: Number of *lemmatized* phrases selected by at least n out of 10 workers.

C. Descriptive Statistics

The experiment was launched and completed by a total of 226 workers, for a mean of 7.07 documents per worker. More than 75% of the workers were based in one of four countries, namely Egypt, Algeria, Saudi Arabia and Tunisia. Only 2.2% of the workers came from countries where Arabic is not an official language, i.e., Germany, Indonesia, Netherlands, France, and Turkey. The time spent reading a document had an average of 302 seconds (5 minutes) and a median of 222 seconds (less than 4 minutes). We collected a total of 10'646 distinct KPs. Figure 1 shows the frequency of the lemmatized phrases, when selected by at least n workers. It is noteworthy that some phrases, when lemmatized, have a frequency higher than 20.

IV. The Arabic Keyphrase Collection

A. Selecting High Quality Keyphrases

Taking all the data as is makes the average number of KPs dramatically high. For example, the SEMEVAL 2010 collection provides an average of only 14.81 KP per document, while if we took all the crowd collected phrases without any kind of filtering, we would end up with a big low-quality dataset including badly selected text, invalid KPs such as stopwords, verbs, etc.

To generate a high-quality collection, we reduced the total number of KPs using selection criteria. Considering only pure KPs (i.e., without diacritics and forbidden symbols) leads to a lower total number of 10'602; if we further apply lemmatization we obtain

Table II: Number of KPs which have been selected by at least n crowd workers for each document.

n Worker	Median	Mean	Min	Max
2:	18	17.8	10	24
4:	4	4.1	1	10
6:	1	1.5	0	4
8:	0	0.5	0	2
10:	0	0.1	0	1
Linguistic cut:	15	15.5	15	19
SEMEVAL:	14	15.1	8	37

10'286, for an average of 64.2 lemmatized phrases per document. To improve the collection quality, we adopted two additional selection approaches:

- *Frequentist*: we order KPs by the number of times that they have been selected by workers, then we discard all the KPs that have not been selected at least twice.
- *Linguistic*: we build a language model and sort the KPs using that model; then, we keep the best 15 ranked phrases per document and discard the others. Note that we keep all phrases that are at the 15^{th} position of the raking, so the actual number of KPs per document will be variable.

Looking at Figure 1, we can see that almost any document has at least 15 KPs selected by at least two workers and a phrase that has been selected by at least 5 workers. Table II offers a more detailed analysis of these data. We see that some documents will have very high-quality KPs, as they were selected by at least 8 workers. Table II also shows that the number of KPs selected by at least $n \geq 2$ workers are pretty similar to the SEMEVAL 2010 dataset.

The linguistic approach deals with phrases which are substrings of other KPs. We built a language model (LM) for each document for a total of 160 LMs. The corpus of each LM consisted in the set of the crowd-assigned KPs and its features are the n-grams (with $n = 1...5$) generated by these KPs. For each LM, we calculated the sum of the columns of the relative frequency table, obtaining a score for each feature, which was dependent from how many times that feature occurred as a (or as part of a) worker selected KP. Then, we excluded the features which were not crowd-assigned KPs, and ranked them by score, obtaining our final linguistic ranking.

The difference between the two approaches is that the language model favors conceptually similar phrases. For example, suppose we have three phrases selected from a document: "computer science teacher", "science teacher" and a completely unrelated word, which is most probably an error, e.g., "foo". Assuming that the first word has been selected three times, while the second and the third one have been selected once, "science teacher" and "foo" have exactly the same score and will be discarded by the frequentist model. The language model, instead, is able to say that "science teacher" is similar to a more frequent KP, pushing it higher in the ranking.

Obviously, each model has its pros and cons; the LM may promote phrases which are actually not important, while the frequentist model may suffer from worker error. Nevertheless, we claim that they are both good ways of ranking worker selected KPs, so we release both rankings in the final dataset.

B. Data Validation

To validate our approaches, we selected a subset of 56 documents from the corpus and had an expert (an Arabic native speaker doing a PhD on KPE) manually assess the quality of the KPs that the crowd selected. The expert was shown the KPs in random order to avoid any bias.

Since we have ranked KPs, it is natural to use classical "top-heavy" ranking metrics, well studied in the IR community. In particular we use the classical ones: Average Precision (AP) and Mean AP (MAP), as well as MAP@5, MAP@10, and MAP@15 to show the quality of the first ranked KPs. Indeed, we show boxplots of AP values (i.e., a representation of the distribution of AP values over the various documents — one dot, one document) to see the variability over documents.

Figure 2 shows that, for all documents and for both linguistic and frequentist approaches, AP@5 and AP@10 values are very high: median values are around 0.9 or higher. AP@15 values are similarly high. MAP@5, MAP@10, and MAP@15 (i.e., the mean AP values, represented as red dots in the figure), are all above 0.8. This means that the sets of the first 5, 10, or 15 KPs in both ranks (i.e., linguistic and frequentist) are very good and sometimes almost perfect. The leftmost boxplot pair in figure shows that AP and MAP values are slightly lower but still well above 0.6: although some of the KPs are not good ones according to our expert, even when using all of them we get a reasonable quality.

We are confident that, when compared with the quality of the other similar KP datasets in the literature, our dataset is at least as reliable. For example, the authors of the SEMEVAL 2010 dataset recognize that only 85% and 81% of their reader- and author-assigned KPs, actually appear in the text and, in contrast to our approach, they simply trust that their readers assigned correct KPs, without using expert knowledge like we do.

C. Applying a Baseline KPE System on the Dataset

Various KPE systems employ TF-IDF as a numerical and statistical method to extract and rank KPs to measure the importance of keywords to a document in a dataset or corpus [4], [3], [7]. Therefore, an Arabic TF-IDF based testbed system was implemented as a baseline KPE to evaluate the quality of the dataset KPs and assess workers performance.

For each document, two lists of words have been generated. The first list contains words of all KPs

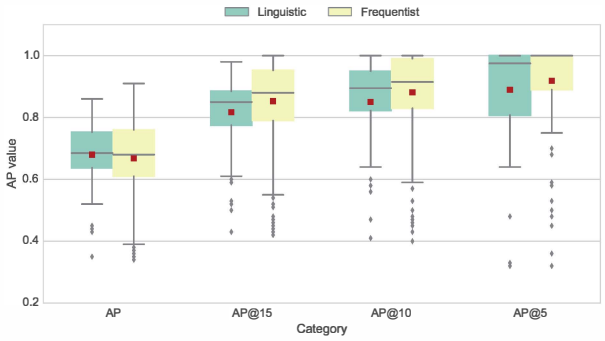

Figure 2: AP of the two approaches at different cuts.

extracted by the workers excluding stopwords while the second one is a sorted list of the important words generated and ranked by the testbed system. After that, the two lists were compared and the precision of the dataset was calculated. The precision was about 0.6 which means that 60% of workers KPs words are recognized by the system; we take this as a good result, especially if compared with the scores of the aforementioned KP extraction systems which rely on this feature.

V. Conclusion and Future Work

We reported on our first effort in building a new KP dataset for Arabic documents by means of crowdsourcing. Being our first effort in building such a corpus, there is plenty of directions to explore in the future. It is possible that we will enlarge the corpus by including more documents; before doing so, however, we intend to study in more detail some issues. For example, we intend to try different approaches and variants to filter the high quality KPs besides those presented in Section IV-A. It will also be important to understand which is the ideal number of workers per document; we have used 10 in our experiment, and a first research direction may be to see if some sampling technique can lead to accurate KPs with lower numbers and, thus, lower cost.

Finally, on a related note, we also plan to try different experimental designs. For instance it would be interesting to try an approach similar to the well known ESP game [16], including the mechanism of taboo words to avoid the crowd to repeatedly select already known KPs. The dataset is available at https://github.com/ailab-uniud/akec, and is structured as follows: 100 randomly selected documents to be used as the training set, and 60 documents as test set. For both sets, for each document, we provide a list of all KPs selected by the workers, randomly ordered, and the two lists of good quality KPs.

References

[1] H. P. Luhn, "Key word-in-context index for technical literature (kwic index)," *American Documentation*, vol. 11, no. 4, pp. 288–295, 1960.

[2] P. D. Turney, "Learning algorithms for keyphrase extraction," *Information Retrieval*, vol. 2, no. 4, pp. 303–336, 2000.

[3] L. Marujo, A. Gershman, J. Carbonell, R. Frederking, and J. P. Neto, "Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization," in *The eighth international conference on Language Resources and Evaluation (LREC)*. ELRA, 2012.

[4] S. R. El-Beltagy and A. Rafea, "Kp-miner: A keyphrase extraction system for english and arabic documents," *Information Systems*, vol. 34, no. 1, pp. 132–144, 2009.

[5] S. N. Kim, O. Medelyan, M.-Y. Kan, and T. Baldwin, "Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles," in *the 5th International Workshop on Semantic Evaluation*. Association for Computational Linguistics (ACL), 2010.

[6] R. Mihalcea and P. Tarau, "Textrank: Bringing order into texts," in *The 42nd Annual Meeting of ACL*. ACL, 2004.

[7] T. D. Nguyen and M.-Y. Kan, "Keyphrase extraction in scientific publications," in *International Conference on Asian Digital Libraries*. Springer, 2007.

[8] A. Awajan, "Keyword extraction from arabic documents using term equivalence classes," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 14, no. 2, p. 7:1, 2015.

[9] J. Chuang, C. D. Manning, and J. Heer, ""without the clutter of unimportant words": Descriptive keyphrases for text visualization," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 19, no. 3, pp. 19:1–19:29, 2012.

[10] W. Zaghouani and K. Dukes, "Can crowdsourcing be used for effective annotation of arabic?" in *LREC*. ELRA, 2014.

[11] M. El-Haj, U. Kruschwitz, and C. Fox, "Creating language resources for under-resourced languages: methodologies, and experiments with arabic," *Language Resources and Evaluation*, vol. 49, no. 3, pp. 549–580, 2015.

[12] A. Al-Thubaity, M. Khan, M. Al-Mazrua, and M. Al-Mousa, "New language resources for arabic: corpus containing more than two million words and a corpus processing tool," in *International Conference on Asian Language Processing (IALP)*. IEEE, 2013.

[13] L. Al-Sulaiti and E. S. Atwell, "The design of a corpus of contemporary arabic," *International Journal of Corpus Linguistics*, vol. 11, no. 2, pp. 135–171, 2006.

[14] M. El-Haj, U. Kruschwitz, and C. Fox, "Using mechanical turk to create a corpus of arabic summaries," in *LREC*. ELRA, 2010.

[15] M. K. Saad and W. Ashour, "Osac: Open source arabic corpora," in *the 6th International Symposium on Electrical and Electronics Engineering and Computer Science (EEECS)*, 2010.

[16] L. Von Ahn and L. Dabbish, "Labeling images with a computer game," in *the SIGCHI conference on Human factors in computing systems*. ACM, 2004.