Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2018) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

The 4th International Conference on Arabic Computational Linguistics (ACLing 2018),
November 17-19 2018, Dubai, United Arab Emirates

Applying Deep Learning for Arabic Keyphrase Extraction

Muhammad Helmy*, R. M. Vigneshram, Giuseppe Serra*, Carlo Tasso

Artificial Intelligence Laboratory, Department of Mathematics, Computer Science, and Physics, University of Udine, Udine 33100, Italy

Abstract

Arabic keyphrase extraction is a crucial task due to the significant and growing amount of Arabic text on the web generated by a huge population. It is becoming a challenge for the community of Arabic natural language processing because of the severe shortage of resources and published processing systems. In this paper we propose a deep learning based approach for Arabic keyphrase extraction that achieves better performance compared to the related competitive approaches. We also introduce the community with an annotated large-scale dataset of about 6000 scientific abstracts which can be used for training, validating and evaluating deep learning approaches for Arabic keyphrase extraction.

Keywords: Arabic NLP; Keyphrase Extraction; Deep Learning

1. Introduction

A keyphrase (KP) is a phrase composed of one or more words (usually up to five) that manifest a main idea or topic of a natural language text document [41]. The objective of any automatic keyphrase extraction (KPE) mechanism is to compile a condensed list of high quality KPs for a given document.

Considering that a massive amount of text documents is produced daily, KPE received more attention as a supportive task in different fields of Natural Language Processing (NLP), information retrieval, document clustering, data-mining, text summarization, and text classification [19, 15, 11].

Typically, KPE systems have two sequential phases: candidate KPs identification, then candidate KPs ranking and selection. In the candidate KPs identification, a set of potential KPs is extracted from the text according to some morphological, syntactical [5] and spatial features. After that, every candidate KP is assigned a score, which reflects its expressiveness according to statistical and semantic measures of its document. Finally, the top KPs are selected. Many paradigms have been developed to tackle KPE task including machine learning , and graph based methods

[*] Corresponding author. Tel.: +39-351-2482124 ; fax: +39-432-558499.
E-mail address: alameldien.muhammad@spes.uniud.it

[42, 40, 7]. These systems need the features for ranking and selecting KPs to be predetermined before running the system. These features cannot be learnt or modified during the system lifetime. It is difficult to enumerate all of the associated features for a specific domain as many linguistic, statistical, and external knowledge about the text should be exploited.

Deep learning (DL) [29] techniques introduced promising approaches for various NLP tasks, which do not require predetermined features. Since Deep Learning approaches require huge datasets to train its models, KPE approaches based on this methodology [31, 6] are designed and developed for English text.

However, in our planet we have a population of about 400 million Arabic native speakers in 28 countries and about 1.7 billion Muslim who are using Arabic as ritual language. Therefore, Arabic KPE is a crucial task due to revolution in the Arabic digital content, especially after Arab Spring, and the emerging needs for annotating and classifying this content.

Moreover, Arabic language has specific characteristics [14], which do not exist in western languages like English, and should be considered when the Arabic text is being processed. These characteristics include diglossia, agglutinative nature, nonconcatenative morphological, ambiguity and high polysemy [18]. To the best of our knowledge, no DL-based approach targeting Arabic KPE task has been reported.

To fill this gap, in this paper we introduce a large-scale dataset suitable for training and testing Arabic KPE models especially DL approaches. We believe that this is the first work which introduce such a dataset. We then propose a DL approach for extracting high quality KPs from Arabic text. The architecture is based on Bidirectional Long Short-Term Memory (Bi-LSTM) Recurrent Neural Network, which is able to exploit previous and future context of a given word in text. The experimental Evaluation shows that the proposed approach outperforms competitive methods. The dataset is available[1] to share them with the community so they can take the opportunity to develop new DL strategies for this task.

2. Related Work

Deep Learning achieved good performance in various NLP tasks including, but not limited to, Language modeling [25], Automatic machine translation (AMT) [12, 17], Named Entity Recognition [28], Sentiment analysis [16], Question answering [37], and lately KPE [44, 6, 31].

DL has been recently taken into consideration for KPE. As mentioned before, The presented DL approaches are devoted for English language, because of abundance of resources created for developing DL systems, e.g., training datasets and word embeddings.

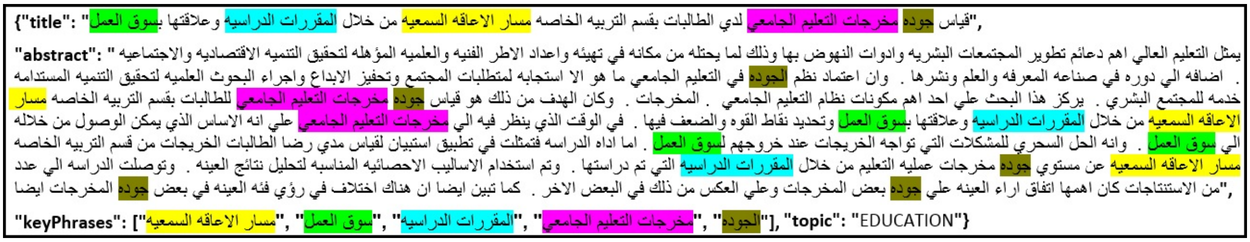
Zhang et al. [44] proposed a novel deep recurrent neural network (RNN) model to tackle the problem of extracting important KPs from tweets where the length restrictions of Twitter-like sites make the performance of existing KPE systems decrease clearly. Tweet length is about 140 characters where the mean size of documents in most of the KPE datasets is more than 500 words [27, 20]. In addition, a huge dataset of tweets was constructed to evaluate the proposed approach.

Basaldella et al. [6] introduced proposed a Deep Long-Short Term Memory Neural Network approach to extract KPs from scientific documents. Since the system does not require hand-craft features dedicated for a specific field, it can be utilized in a wide range of domains. The system has been evaluated on INSPEC dataset [22].

DL was utilized in KP generation and assignment, where the system can recognize KPs that do not exist in the document, and takes into account the actual semantic meaning behind the text. Meng et al. [31] introduced a generative model (deep keyphrase generation) for KP prediction with an encoder-decoder framework. The authors built a dataset consists of 20,000 scientific documents to train and evaluate the system.

For sake of completeness, Deep Learning has been also employed in few areas in Arabic NLP like text categorization [23], sentiment analysis [4], and question answering [38]. As far as we know, there is not a system which employs DL approaches in KPE from Arabic text.

[1] http://ailab.uniud.it/arabickpe/



```
{"title": "قياس جوده مخرجات التعليم الجامعي لدي الطالبات بقسم التربيه الخاصه مسار الاعاقه السمعيه من خلال المقررات الدراسيه وعلاقتها بسوق العمل",
"abstract": "يمثل التعليم العالي اهم دعائم تطوير المجتمعات البشريه وادوات النهوض بها وذلك لما يحتله من مكانه في تهيئه واعداد الاطر الفنيه والعلميه المؤهله لتحقيق التنميه الاقتصاديه والاجتماعيه
. اضافه الي دوره في صناعه المعرفه والعلم ونشرها . وان اعتماد نظم الجوده في التعليم الجامعي ما هو الا استجابه لمتطلبات المجتمع وتحفيز الابداع واجراء البحوث العلميه لتحقيق التنميه المستدامه
خدمه للمجتمع البشري . يركز هذا البحث علي احد اهم مكونات نظام التعليم الجامعي . المخرجات . وكان الهدف من ذلك هو قياس جوده مخرجات التعليم الجامعي للطالبات بقسم التربيه الخاصه مسار
الاعاقه السمعيه من خلال المقررات الدراسيه وعلاقتها بسوق العمل وتحديد نقاط القوه والضعف فيها . في الوقت الذي ينظر فيه الي مخرجات التعليم الجامعي علي انه الاساس الذي يمكن الوصول من خلاله
الي سوق العمل . وانه الحل السحري للمشكلات التي تواجه الخريجات عند خروجهم لسوق العمل . اما اداه الدراسه فتمثلت في تطبيق استبيان لقياس مدي رضا الطالبات الخريجات من قسم التربيه الخاصه
مسار الاعاقه السمعيه عن مستوي جوده مخرجات عمليه التعليم من خلال المقررات الدراسيه التي تم دراستها . وتم استخدام الاساليب الاحصائيه المناسبه لتحليل نتائج العينه . وتوصلت الدراسه الي عدد
من الاستنتاجات كان اهمها اتفاق اراء العينه علي جوده بعض المخرجات وعلي العكس من ذلك في البعض الاخر . كما تبين ايضا ان هناك اختلاف في رؤي فئه العينه في بعض جوده المخرجات ايضا",
"keyPhrases": ["مسار الاعاقه السمعيه", "سوق العمل", "المقررات الدراسيه", "مخرجات التعليم الجامعي", "الجوده"], "topic": "EDUCATION"}
```

Fig. 1: An Example of the dataset where each keyphrase is colored to indicate where it exists in the abstract and title.

3. Arabic Keyphrase Dataset

There is not any large-scale dataset for Arabic KPE that can be used to train, validate and test a deep learning model. We found only three small publicly available datasets:

- Arabic Keyphrase Extraction Corpus (AKEC) [20]: The corpus consists of 160 Arabic documents and their assigned KPs. The authors employed the crowdsourcing platform of Crowdflower to construct the collection with the support of 226 workers. AKEC[2] is the first dataset which is not customized or annotated by the authors of the KPE system.
- Arabic Dataset[3] [1]: the dataset contains 400 documents and covers 18 different topics. All of the documents were assigned to six readers only to read and extract 10 KPs for each.
- WikiAll [13]: it is composed of 100 documents collected from Arabic Wikipedia[4]. The average size of document is 804 word and the average number of assigned KPs per document is 8.1. The documents are not preprocessed or organized in categories. Moreover, the metadata of Wikipedia are still there in the documents text.

Since these datasets are fairly small (total number of documents is about 660 document), they can not be used as training datasets. They may be employed as test sets. Therefore, we started to build a large dataset of the scientific articles abstracts written and published in Arabic language.

We targeted web sites of the scientific journals of the Arabic universities and some Arabic literature publishers to crawl the abstracts available freely with their keyphrases, titles, and topics. A set of 6219 abstracts has been crawled. The total number of KPs assigned by authors is 26,685 with 15,730 KP that appear verbatim in text and 10,955 KP do not exist in the abstracts text. Finally, we removed all of the absent KPs and exclude the documents which have no assigned KPs. The total number of documents after preprocessing became about 6000 documents.The total number of words in the dataset text is 1,223,723 word, and the vocabulary size is 68,108 unique word.

The collection of abstracts was arbitrarily split into three sets: a training set (used during building and training the model), contains about 4,000 documents, a validation set (to evaluate the various model cases with different parameters, and select the best performing one), consists of about 1,000 documents, and a test set (to obtain impartial results of different systems) with the remaining 1,000 abstracts. The dataset is stored in JSON format where each item (document) of the dataset contains title, abstract, keyphrases and the topic the item belongs to. Figure 1 shows an example of a dataset item.

Table 1 shows statistics about the dataset. Where Docs refers to the total number of documents in every item. KPs is the total number of KPs verbally exist in the text. Words is the summation of all words within the documents whether it is repeated or not. Vocabulary is the number of unique words i.e. without repetition. Finally, the table presents the maximum, minimum, average, and median value for document size (Doc size), in words, and the number of KPs (No. of KPs) assigned for documents.

[2] https://github.com/ailab-uniud/akec

[3] https://github.com/logmani/ArabicDataset

[4] https://ar.wikipedia.org/wiki/

Table 1: Statistical information of our dataset.

		Training	Validation	Test
Docs		4000	1000	1000
KPs		10582	2583	2565
Words		1026938	195630	196785
Vocabulary		62204	24424	24373
Doc size	Max	994	761	634
	Min	25	45	31
	Avg	210	207.24	209.12
	Med	195	194	194
No. of KPs	Max	11	9	13
	Min	1	1	1
	Avg	2.69	2.74	2.72
	Med	3	3	3

To determine whether our dataset is comparable to the well-established English datasets, we compare the total number of KPs, present KPs, and absent KPs of our dataset against four English datasets. The comparison is presented in Table 2. The four author-assigned keyphrases English datasets are:

- Krapivin [26]: includes about 2,304 high quality documents representing scientific articles from computer science domain. It was dedicated for training and evaluating machine learning-based KPE approaches.
- NUS [35]: consists of 211 conference articles, with a length range of 4-12 pages. The documents were converted into plain text format and originally downloaded using Google SOAP API as PDF documents. Volunteers were recruited to assign KPs to each document which allows multiple judgments beside the author-assigned KPs.
- Inspec [22]: is a collection of 2,000 abstracts, with their corresponding titles and KPs from Inspec[5] which is an indexing database of scientific and technical literature, published by the Institution of Engineering and Technology (IET)[6]. The dataset was randomly divided into three parts: a training set consisting of 1,000 documents, a validation set consisting of 500 documents, and a test set with the remaining 500 abstracts.
- SemEval-2010 [24]: it is composed of 288 documents collected from ACM Digital Library. The dataset was constructed for evaluating participant systems of Task 5 of the Workshop on Semantic Evaluation 2010 (SemEval-2010)[7]. The size of the documents ranges from 6 to 8 pages from a variety of different topics. The collection is divided into three parts: training (144 documents), test (100 documents) and trial (40 documents).

4. The Proposed System

A DL model was developed based on LSTM. We utilized an existing general purpose Arabic word embeddings for training the model. The description of the system components will be discussed in the following subsections.

4.1. Word Embeddings

Word embedding simply map the words or phrases of natural text into vectors of real numbers. The main two approaches available for building word embeddings from raw text are: GloVe [36] and word2vec model [32]. Word2vec, in turn, has two approaches for computing the word vectors, the skip-gram which predicts the context-words from a given source word, and Continuous Bag-Of-Words (CBOW) which predicts a word given its context window [33].

[5] https://inspecdirect.theiet.org/

[6] https://www.theiet.org/

[7] http://semeval2.fbk.eu/semeval2.php?location=tasks#T6

Table 2: Comparison of present and absent keyphrases percentage: four English datasets [31] and our dataset.

Dataset	#Keyphrase	% Present	% Absent
Inspec	19,275	55.69	44.31
Krapivin	2,461	44.74	55.26
NUS	2,834	67.75	32.25
SemEval	12,296	42.01	57.99
Our dataset	26,685	58.95	41.05

Table 3: Forms of Arabic text during preprocessing.

Type	Text
Original	رَكِـبْتُ سَـــيَاْرَتِي إِلَى مَـــدِينَةِ الْقَــاهِـرَةِ
Trans	I drove my car to Cairo city
No Diac	ركـــبت ســـيارتي إلى مـــدينة القــاهـرة
Normal	ركبت سيارتي الي مدينه القاهره
Segmented	ركب ت سياره ي الي مدينه القاهره

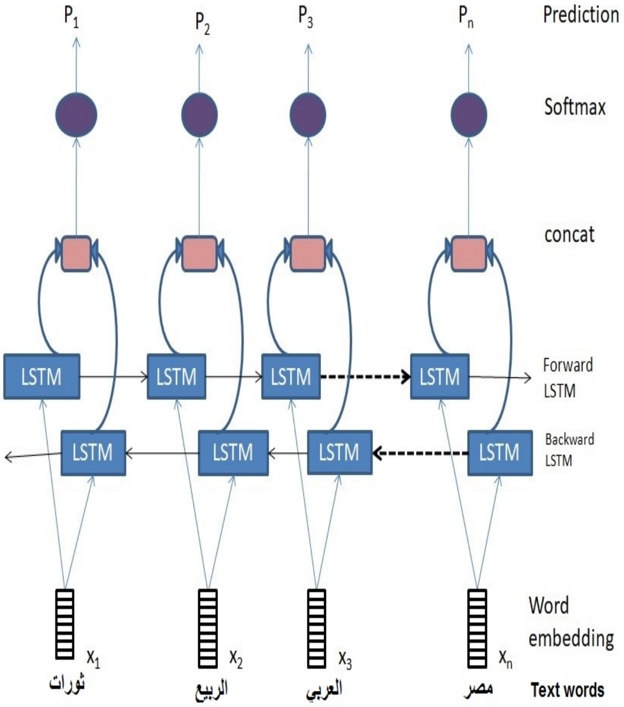


Fig. 2: Model architecture

All NLP researchers of the Arabic DL systems build customized Word embedding for their applications and most of them are not published [9, 3, 2]. However, we found two public global word embedding sets; the first one uses Glove and Word2Vec with vector size of 300 only [43], the second one is called AraVec and uses the Word2Vec approach with three different vector sizes (e.g. 300, 100, and 50) [39]. We decided to use AraVec for our system since the first one includes bigram phrases which are not required in our pipeline.

4.2. Model Architecture

KPE is performed by the following procedure: the document text is preprocessed to represent the text in the form of separated words. Preprocessing Arabic text includes removing Arabic diacritics (which represent short vowels and consonant), normalizing different shapes of Arabic characters into a single shape (i.e Alef letter has different shapes: أ، إ، آ which is normalized to ا), finally, segmenting the text into single tokens (Arabic word may contain more than one token or word) using Stanford CoreNLP Toolkit [30]. Table 3 shows different forms of Arabic text during preprocessing. Then, the documents are divided into sentences and associate the tokens with the word embedding representation.

Let the input tokens of word embedding represented as $\{x_1, ..., x_n\}$, a Recurrent Neural Network (RNN) determines the output vector of each token by iteration.

The embedding layer works as a lookup table that transforms discrete features such as the words of Arabic text into continuous real-valued vector representations, which are then concatenated and provided to the neural network. Instead of a feed-forward network, we utilize the bi-directional long-short term memory (BLSTM) network.

KPE can be considered as sequential labeling task which involves the algorithmic assignment of a categorical label to each member of a sequence of observed values. In such task, a bi-directional LSTM model can take into consideration an adequately enormous amount of context on both sides of a word and erase limited context problem that applies to any feed-forward model.

Table 4: Comparison results on our dataset.

	Top 5 KPs			Top 10 KPs			Top 15 KPs			
Approach	P	R	F1	P	R	F1	P	R	F1	MAP
TF-IDF	0.170	0.255	0.204	0.085	0.256	0.128	0.057	0.256	0.093	0.255
KP-Miner	0.201	0.303	0.242	0.101	0.304	0.151	0.067	0.305	0.110	0.303
Our System	**0.305**	**0.444**	**0.361**	**0.208**	**0.588**	**0.308**	**0.160**	**0.671**	**0.258**	**0.471**

The Bidirectional LSTM network adopts the future context. In fact, with this architecture we are able to make use of both past context and future context of a specific word. It consists of two separate hidden layers; it computes the forward hidden sequence then, it computes the backward hidden sequence and finally, it combines forward hidden sequence and backward hidden sequence to generate the output. The combination (concat) layer is connected to a softmax output layer with three neurons for each word. The three neurons are associated with three possible output classes, which respectively mark tokens that are not keyphrases, the first token of a keyphrase, and the internal tokens of a keyphrase. A dropout technique was implemented between Bi-LSTM and the dense layer to prevent overfitting. Figure 2 shows the basic structure of the model.

4.3. Model Implementation

We have used Keras[8] with Tensorflow[9] as a backend. That in turn allowed us to employ CUDA[10] to train our neural network using GPU framework (GeForce GTX 1080 Ti Graphics Card)[11]. After trying different configurations for the network, we obtained the best results with a size of 150 neurons for the Bi-LSTM layer, 150 neurons for the hidden dense layer, and a value of 0.25 for the dropout layers. During the training of our network, we used Root Mean Square Propagation optimization algorithm and batch size of 32. The early stopping rule in Keras on embedding is used to terminate the training process when the training loss does not decrease for two consecutive epochs.

5. Evaluation and Experimental Results

The evaluation experiments were conducted on two datasets, our test dataset and WikiAll dataset [13]. We choose WikiAll as an evaluation dataset, because it has been used by most of the published Arabic KPE systems.

The first experiment was carried out using our test dataset. We compare the performance of our system against two available published systems: The first one is Distiller TF-IDF (D-TF-IDF) [8] which is a pipeline implemented within the Distiller framework for extracting KPs using the simple statistical approach of Term FrequencyInverse Document Frequency (TF-IDF). Distiller [10] is a knowledge extraction framework which provide a flexible, multilingual KPE functionalities for about five languages, one of them is Arabic. The second system is KP-Miner [13] which is based on an unsupervised approach for KPE. It does not need to be trained on a particular document set in order to achieve its task. KP-Miner can extract KPs from a single document or a corpus of documents. Its heuristic rules can be configured to suit the document domain and user understandings of the document nature.

We check the systems performance over the top 5, 10 and 15 candidates KPs returned by each system. The lemmatized versions of the returned KPs are matched with the lemmatized KPs assigned to the dataset documents. Then, we calculated the Precision (P), Recall (R), F1-score (F1), and Mean Average Precision (MAP) as evaluation metrics. Table 4 shows these comparison results where our system achieves higher performance.

The second experiment was conducted on WikiAll dataset. We compare the performance in term of Precision, Recall and Average number of correct detected KPs (Avg. Keys) which are used by the competitive systems. Table 5 shows the performances of five different approaches and our approach. The five approaches are KP-Miner [13], Arabic TF-IDF, Word2Vec, Hyprid model [34], and MorphKE [21].

[8] https://keras.io/
[9] https://www.tensorflow.org/
[10] https://developer.nvidia.com/cuda-toolkit
[11] https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx-1080-ti/

Table 5: Performance results of our approach compared to other published results on WikiAll dataset.

	Precision	Recall	Avg. Keys
KP-Miner	0.131	0.383	2.491
TF-IDF	0.112	0.349	2.253
Word2Vec	0.092	0.294	1.701
Hybrid	0.101	0.312	2.002
MorphKE	0.131	0.377	2.530
Our System	**0.216**	**0.419**	**2.851**

In Arabic TF-IDF, The candidate KPs are weighted and scored using TFxIDF algorithm which gives low weights to the unimportant KPs. In addition, it uses a list of stopwords which is very beneficial for Arabic text as some stop-words in Arabic are compound ones and do not occur frequently. Word2Vec approach employed Googles Word2Vec[12] library to measure the similarity between the candidate patterns and the document title. The System was trained using Wikipedia Arabic dump[13] to get the vector representation of the words, then the cosine similarity was used to measure the distance between the title of each document and its valid KPs patterns. The hybrid approach is a combination model of Arabic TFxIDF and Word2Vec models [34]. MorphKE is an unsupervised approach based on utilizing the rich Arabic morphology and syntax to generate a restricted set of meaningful candidates KPs for a single document [21]. The experimental results showed that the proposed approach performs significantly better than previous methods.

6. Conclusion

In this article, we introduced a deep learning KPE approach based on the Bi-LSTM neural network model for extracting keyphrases from Arabic text. Since we have a shortage in large-scale datasets for training and evaluating deep learning models for Arabic KPE, we construct a new dataset consisting in about 6,000 abstracts of scientific Arabic documents. The dataset attributes are comparable to the English datasets. We used the dataset to train, validate, and test our approach against the existing systems. The evaluation results show that our approach achieves state-of-the-art performance in Arabic KPE domain.

References

[1] Al Logmani, M., Al Muhtaseb, H., 2017. Arabic dataset for automatic keyphrase extraction, in: International Conference on Computer Science, Information Technology and Applications, pp. 217–222.

[2] Al-Sallab, A., Baly, R., Hajj, H., Shaban, K.B., El-Hajj, W., Badaro, G., 2017. Aroma: A recursive deep learning model for opinion mining in arabic as a low resource language. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP) 16, 1–20.

[3] Altowayan, A.A., Tao, L., 2016. Word embeddings for arabic sentiment analysis, in: IEEE International Conference on Big Data, pp. 3820–3825.

[4] Badaro, G., Baly, R., Hajj, H., Habash, N., El-Hajj, W., 2014. A large scale arabic sentiment lexicon for arabic opinion mining, in: Conference on Empirical methods in Natural Language Processing (EMNLP) Workshop on Arabic Natural Language Processing (ANLP), pp. 165–173.

[5] Barker, K., Cornacchia, N., 2000. Using noun phrase heads to extract document keyphrases, in: Conference of the Canadian Society for Computational Studies of Intelligence, pp. 40–52.

[6] Basaldella, M., Antolli, E., Serra, G., Tasso, C., 2018a. Bidirectional lstm recurrent neural network for keyphrase extraction, in: Italian Research Conference on Digital Libraries (IRCDL), pp. 180–187.

[7] Basaldella, M., Helmy, M., Antolli, E., Popescu, M.H., Serra, G., Tasso, C., 2017. Exploiting and evaluating a supervised, multilanguage keyphrase extraction pipeline for under-resourced languages, in: International Conference Recent Advances in Natural Language Processing (RANLP), pp. 78–85.

[8] Basaldella, M., Serra, G., Tasso, C., 2018b. The distiller framework: Current state and future challenges, in: IRCDL, pp. 93–100.

[9] Dahou, A., Xiong, S., Zhou, J., Haddoud, M.H., Duan, P., 2016. Word embeddings and convolutional neural network for arabic sentiment classification, in: International Conference on Computational Linguistics, pp. 2418–2427.

[12] https://code.google.com/p/word2vec/

[13] https://github.com/anastaw/Arabic-Wikipedia-Corpus

[10] De Nart, D., Degl'Innocenti, D., Tasso, C., 2015. Introducing distiller: a lightweight framework for knowledge extraction and filtering, in: The 23rd Conference on User Modelling, Adaptation and Personalization (UMAP).

[11] Degl'Innocenti, D., De Nart, D., Helmy, M., Tasso, C., 2018. Fast, accurate, multilingual semantic relatedness measurement using wikipedia links, in: Intelligent Natural Language Processing: Trends and Applications, pp. 571–584.

[12] Deselaers, T., Hasan, S., Bender, O., Ney, H., 2009. A deep learning approach to machine transliteration, in: Association for Computational Linguistics (ACL) Workshop on Statistical Machine Translation, pp. 233–241.

[13] El-Beltagy, S.R., Rafea, A., 2009. Kp-miner: A keyphrase extraction system for english and arabic documents. Information Systems 34, 132–144.

[14] Farghaly, A., Shaalan, K., 2009. Arabic natural language processing: Challenges and solutions. TALLIP 8, 14:1–14:22.

[15] Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-Manning, C.G., 1999. Domain-specific keyphrase extraction, in: International Joint Conference on Artificial Intelligence, pp. 668–673.

[16] Glorot, X., Bordes, A., Bengio, Y., 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach, in: International Conference on Machine Learning (ICML), pp. 513–520.

[17] Guzmán, F., Bouamor, H., Baly, R., Habash, N., 2016. Machine translation evaluation for arabic using morphologically-enriched embeddings, in: International Conference on Computational Linguistics, pp. 1398–1408.

[18] Habash, N.Y., 2010. Introduction to arabic natural language processing. Synthesis Lectures on Human Language Technologies 3, 1–187.

[19] Hasan, K.S., Ng, V., 2014. Automatic keyphrase extraction: A survey of the state of the art, in: ACL, pp. 1262–1273.

[20] Helmy, M., Basaldella, M., Maddalena, E., Mizzaro, S., Demartini, G., 2016a. Towards building a standard dataset for arabic keyphrase extraction evaluation, in: 20th International Conference on Asian Language Processing (IALP), pp. 26–29.

[21] Helmy, M., De Nart, D., Degl'Innocenti, D., Tasso, C., 2016b. Leveraging arabic morphology and syntax for achieving better keyphrase extraction, in: 20th International Conference on Asian Language Processing (IALP), pp. 340–343.

[22] Hulth, A., 2003. Improved automatic keyword extraction given more linguistic knowledge, in: EMNLP, pp. 216–223.

[23] Jindal, V., 2016. A personalized markov clustering and deep learning approach for arabic text categorization, in: ACL Student Research Workshop, pp. 145–151.

[24] Kim, S.N., Medelyan, O., Kan, M.Y., Baldwin, T., 2010. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles, in: ACL Workshop on Semantic Evaluation, pp. 21–26.

[25] Kim, Y., Jernite, Y., Sontag, D., Rush, A.M., 2016. Character-aware neural language models., in: Association for the Advancement of Artificial Intelligence, pp. 2741–2749.

[26] Krapivin, M., Autaeu, A., Marchese, M., 2009. Large dataset for keyphrases extraction. Technical Report. University of Trento.

[27] Krapivin, M., Autayeu, M., Marchese, M., Blanzieri, E., Segata, N., 2010. Improving machine learning approaches for keyphrases extraction from scientific documents with natural language knowledge, in: the joint JCDL/ICADL international digital libraries conference, pp. 102–111.

[28] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C., 2016. Neural architectures for named entity recognition, in: Conference of the North American Chapter of the Association for Computational Linguistics (NAACL): Human Language Technologies, pp. 260–270.

[29] LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.

[30] Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D., 2014. The Stanford CoreNLP natural language processing toolkit, in: ACL, System Demonstrations, pp. 55–60.

[31] Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., Chi, Y., 2017. Deep keyphrase generation, in: ACL, pp. 582–592.

[32] Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013a. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 .

[33] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013b. Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, pp. 3111–3119.

[34] Nabil, M., Atiya, A., Aly, M., 2015. New approaches for extracting arabic keyphrases, in: International Conference on Arabic Computational Linguistics, pp. 133–137.

[35] Nguyen, T.D., Kan, M.Y., 2007. Keyphrase extraction in scientific publications, in: International Conference on Asian Digital Libraries, pp. 317–326.

[36] Pennington, J., Socher, R., Manning, C., 2014. Glove: Global vectors for word representation, in: EMNLP, pp. 1532–1543.

[37] Qiu, X., Huang, X., 2015. Convolutional neural tensor network architecture for community-based question answering., in: International Joint Conferences on Artificial Intelligence (IJCAI), pp. 1305–1311.

[38] Romeo, S., Da San Martino, G., Belinkov, Y., Barrón-Cedeño, A., Eldesouki, M., Darwish, K., Mubarak, H., Glass, J., Moschitti, A., 2017. Language processing and learning models for community question answering in arabic. Information Processing & Management .

[39] Soliman, A.B., Eissa, K., El-Beltagy, S.R., 2017. Aravec: A set of arabic word embedding models for use in arabic nlp. Procedia Computer Science 117, 256–265.

[40] Tixier, A., Malliaros, F., Vazirgiannis, M., 2016. A graph degeneracy-based approach to keyword extraction, in: EMNLP, pp. 1860–1870.

[41] Turney, P.D., 2000. Learning algorithms for keyphrase extraction. Information retrieval 2, 303–336.

[42] Ying, Y., Qingping, T., Qinzheng, X., Ping, Z., Panpan, L., 2017. A graph-based approach of automatic keyphrase extraction. Procedia Computer Science 107, 248–255.

[43] Zahran, M.A., Magooda, A., Mahgoub, A.Y., Raafat, H., Rashwan, M., Atyia, A., 2015. Word representations in vector space and their applications for arabic, in: International Conference on Intelligent Text Processing and Computational Linguistics, pp. 430–443.

[44] Zhang, Q., Wang, Y., Gong, Y., Huang, X., 2016. Keyphrase extraction using deep recurrent neural networks on twitter, in: EMNLP, pp. 836–845.