A Data-Driven Approach to Automatic Extraction of Professional Figure Profiles from Résumés

Alessandro Bondielli[1,2](✉) and Francesco Marcelloni[2]

[1] DINFO, University of Florence, Florence, Italy
alessandro.bondielli@unifi.it
[2] Department of Information Engineering, University of Pisa, Pisa, Italy
francesco.marcelloni@unipi.it

Abstract. The process of selecting and interviewing suitable candidates for a job position is time-consuming and labour-intensive. Despite the existence of software applications aimed at helping professional recruiters in the process, only recently with Industry 4.0 there has been a real interest in implementing autonomous and data-driven approaches that can provide insights and practical assistance to recruiters.

In this paper, we propose a framework that is aimed at improving the performances of an *Applicant Tracking System*. More specifically, we exploit advanced Natural Language Processing and Text Mining techniques to automatically profile resources (i.e. candidates for a job) and offers by extracting relevant keywords and building a semantic representation of résumés and job opportunities.

1 Introduction

For any company, it is of paramount importance to identify the right person for a job, especially in today's very dynamic market, where new technologies and specific professional figures considerably increase or decrease in popularity in a very short time span. Thus, it is crucial, especially in the context of Industry 4.0, to have data-driven tools which extract autonomously profiles of professional figures from both job opportunities and résumés, so as to capture the appearance of novel required competences and knowledge. Indeed, given the enormous amount of information and profiles on professional-oriented online platforms, such as *Linkedin*[1], these tools may help recruiters to identify the most suitable candidates for a given job offer. Further, by analyzing regularities and variations among data, they may allow predicting upcoming trends in the job market and identifying emerging and yet unseen professional figures.

Most of the current recruitment support approaches fall under the umbrella of so called *Applicant Tracking Systems* (ATSs). Available ATSs are comparable to Customer Relationship Management systems, with a focus on supporting

[1] https://www.linkedin.com.

© Springer Nature Switzerland AG 2019

H. Yin et al. (Eds.): IDEAL 2019, LNCS 11871, pp. 155–165, 2019.
https://doi.org/10.1007/978-3-030-33607-3_18

discovery and selection of candidates for job opportunities. Such systems are often based on manual collection and annotation of résumés and job opportunities by experienced recruiters, and on rule-based algorithms to identify suitable candidates.

In this paper, we present a preliminary version of a data-driven approach to the problem of automatically generating profiles of professional figures, by extracting relevant information from résumés and job opportunities. A classic and straightforward document similarity calculation approach cannot guarantee satisfactory performance in the ATS domain. This is due to the fact that documents in this domain are mostly similar to each other, and tend to be differentiated by minor aspects. For example, the two sentences "fluent in Python" and "fluent in Java" are extremely similar, but describe two different professional figures. Therefore, a simple document similarity calculation on whole documents could be unsuccessful since it may be prone to simply identify similarly written curricula. For this reason, we employ a composite approach which exploits Natural Language Processing (NLP) techniques, such as *word entropy* [4], for identifying relevant keywords, and recent distributional semantics techniques, such as word and document vectors [11], in order to represent similar profiles. Further, our approach exploits only the unstructured information contained in résumés. To the best of our knowledge, current ATSs do not support this kind of analysis and require a frequent interaction with the human experts. On the other hand, the development of ATSs has a distinctly commercial nature. Thus, the research community has not been particularly active on this specific topic. Today, with Industry 4.0, there is a growing interest in implementing autonomous and data-driven approaches that can provide insights on how the job market evolves over time, and practical assistance to recruiters. This is definitely a novelty for most commercial ATSs, especially for those targeted to small-to-medium sized companies that focus on a small domain of application.

This paper is the result of a collaboration with the recruitment agency IT Partner Italia[2]. The company mostly operates in the field of IT system integration. Due to the high volume of hiring processes managed on a daily basis, IT Partner Italia is aiming to build a next-generation ATS based, among other technologies, on NLP and Machine Learning techniques, in order to help recruiters in their most time consuming activities.

This paper is organized as follows. In Sect. 2 we discuss related work. Section 3 illustrates our proposed approach. Section 4 presents a preliminary evaluation of the approach on a subset of data. Finally, Sect. 5 draws some conclusion.

2 Related Work

Actually, very few researches on the impact of ATSs on the recruitment process have been proposed within the business economics field [5,10]. In addition, a number of recent approaches have proposed the implementation of data mining

[2] http://www.itpartneritalia.com/.

and information retrieval techniques both in the recruitment process and in performing job recommendations [9,16]. However, it is clear that extensive research in the direction of improving software such as ATS is still in its embryonic stage. Today's ATSs belong to two main categories: *direct recruitment* and *indirect recruitment* ATSs.

The target of the first category is the human resources department of a specific company, in order to directly handle the selection of candidates and the employment process. Thus, these ATSs are often developed to be easily integrated with other internal systems such as Enterprise Resource Planners.

Indirect recruitment ATSs, on the other hand, are developed for consulting, recruiting, and interim agencies. For this reason, such ATSs need to satisfy a wider range of criteria, concerning efficiency, autonomy, user-friendliness and speed of execution.

In the field of NLP, we have been observing an exponential growth of approaches aimed at representing words and documents in a semantically meaningful way. Most of such approaches are based on the observation originally proposed in [8] that words with similar meaning occur in similar contexts. Thus, the first steps along this direction were aimed at building implicit vector representation of words, by means of co-occurrences with other words.

More recently, deep learning has been extensively used for learning word representations [1,12,13]. The study of *word embeddings* has received in fact a lot of attention in the last few years thanks to the effectiveness of such representation in capturing a semantic quality of words. Recent literature has also focused on *character level embeddings* [2], that are able to represent *subword* information as well as word information, and *deep language models* [14], that focus on improving word embeddings models by taking into account contexts in order to model *polisemy* of words. On the contrary, a limited attention has been paid to build frameworks for representing sentences, paragraphs and entire documents embeddings, such as proposed in [11]. However, we believe that such kind of representations would have a great value in industrial environment, where both time and performances are important constraints.

3 The Proposed Approach

We aim to define a method to automatically extract the different profiles of professional figures from available résumés and job opportunities. Our approach consists of two main phases. In the first phase, we use NLP techniques to generate representations of résumés and job opportunities. In the second phase, we exploit these representations for generating profiles by exploiting clustering techniques. An overview of the entire approach is given in Fig. 1. This paper focuses mainly on the description and early evaluation of the first phase. In particular, we extract potential candidate keywords from each résumé and job opportunity by employing entropy as a weighting measure for their relevance. Then, we exploit the extracted keywords to generate a distributional semantic model of résumés and job opportunities by means of the *paragraph vector* algorithm [11].

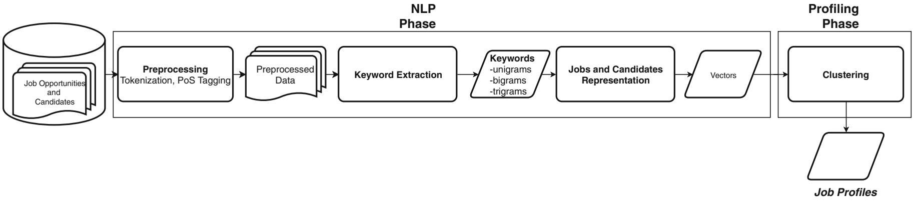
NLP Phase
Profiling Phase
Job Opportunities and Candidates
Preprocessing Tokenization, PoS Tagging
Preprocessed Data
Keyword Extraction
Keywords -unigrams -bigrams -trigrams
Jobs and Candidates Representation
Vectors
Clustering
Job Profiles

Fig. 1. Overview of the proposed approach.

3.1 Preprocessing

In order to improve the quality of our representations and to reduce the computational cost, we perform a preprocessing step. More specifically, we execute *sentence splitting*, *tokenization* and *Part-of-Speech (PoS) Tagging* by using SpaCy[3], a Python toolkit which provides deep learning based models to perform linguistic analysis for several languages, including Italian, out of the box. After tokenization and PoS Tagging, we also remove several stopwords (e.g. "Curriculum Vitae") that are not interesting in our analysis.

3.2 Keyword Extraction

Our keyword extraction algorithm has two main goals. First, we aim to identify keywords that can help us in profiling and describing résumés and job opportunities. Second, we want to be able to take into account the time, i.e. when a given résumé or job opportunity was produced and/or updated. The temporal aspect may in fact be crucial to enable the identification of regularities, i.e. recurrent terms and novelties in word usage. Such novelty in turn might be representative of previously unseen profiles.

Given the context of application, in addition to single words, we believe that *multiword expressions* should be taken into account as well. As in [6], we consider multiword expressions as "lexicalised word combinations as a theoretical, phraseological notion". For example, in our context of application, expressions representing technologies or software, such as "Dynamics AX", "Adobe Photoshop" or "SQL Server", should be considered as single semantic units of texts, i.e. multiword expressions. Such expressions may in fact provide a crucial insight in the process of identifying relevant keywords, as well as in the production of semantically relevant representation of résumés and job opportunities. For the sake of simplicity and readability, we will refer to both single words and multiword expressions as *n-grams*.

The keyword extraction algorithm is quite simple. Figure 2 shows its flowchart. We adopt a window-based strategy in order to identify relevant keywords for a given period of time. The time-window size is in the range $[1, n]$ days.

[3] https://spacy.io/.

For each time window, we compute the frequency of n-grams with respect to each document. We consider n-grams from unigrams to trigrams. In order to reduce complexity and noise, we consider only n-grams with specific PoS patterns. For instance, we want to identify n-grams such as “full stack developer”, but have no interest in n-grams like “expert in”, that would increase the computational complexity of the algorithm without extracting interesting terms. We then use the relative frequencies found in the time window to perform an on-line update of the overall frequency counts, and the weights for each n-gram. Once the weights are recomputed, we extract from current documents all n-grams whose weights are larger than a pre-fixed threshold WT.

This strategy allows us to identify both terms that occur frequently across all time windows, and previously not considered terms that appear as relevant only for a given window. In addition to the set of keywords extracted automatically from the documents, we also employ a set of keywords that have been manually selected as relevant by experienced recruiters. In our framework, both manually selected keywords and entropy-based keywords coexist. More specifically, we aim to introduce human experts in the loop, by enabling an *a-posteriori* selection of keywords considered as *relevant*, among those found by our algorithm, and marking as *noise* the least interesting ones. This is expected to improve performances when running the algorithm on novel data.

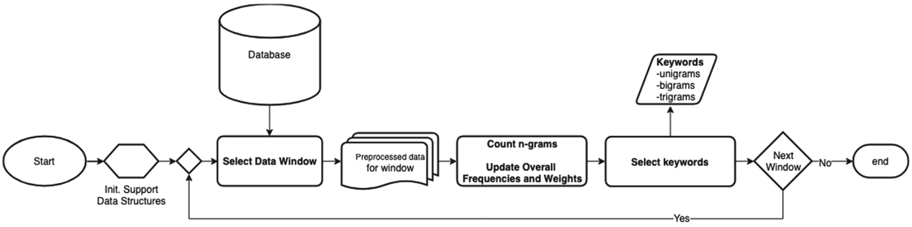

Fig. 2. Flowchart of the keyword extraction algorithm.

Concerning the selection criteria of n-grams, we consider specific content word PoS (e.g. *nouns*, *adjectives*) and specific PoS patterns that are often found in multiword expressions, such as *adjective-adposition-noun*, as in “fluent in Python”. In order to decide whether an n-gram is relevant or not, we exploit classical association measures such as *Pointwise Mutual Information* (PMI) [3]. The PMI of two or more words quantifies the discrepancy between their joint probability and their individual marginal probability by assuming independence. More specifically, given two words x and y,

$$PMI(x;y) = \log \frac{p(x,y)}{p(x) * p(y)}$$

Probabilities are estimated by means of frequency. In particular, $p(x)$ (and $p(y)$) can be easily estimated as $f(x)/N$, where $f(x)$ is the frequency of x in the data, and N the total number of unigrams. Thus, $p(x,y) = f(x,y)/N_{bigrams}$.

In addition, since PMI is based on joint probabilities of events, it satisfies the *chain rule* (or general product rule) of probability. Thus, PMI for trigrams is computed as $PMI(x; yz) = PMI(x; y) + PMI(x; z|y)$.

We implement a slight variation of the PMI formula, namely *Positive PMI* (PPMI). PPMI is simply defined as $PPMI(x; y) = [PMI(x; y)]_+$.

We select as candidate keywords only multi-word terms that have a PPMI equal to or larger than 1.00. This ensures that we only select pairs or triplets of words that have a chance of co-occurring together higher than they were independent. Note that PPMI for unigrams is set as 1.00 by default.

Concerning the weighting of selected n-grams, several approaches for extracting relevant keywords from text documents have been proposed in the literature. We chose to adopt a weighting scheme based on term entropy. Term entropy [15] measures the average uncertainty of a given term in a collection of documents [4], thus providing an efficient method for estimating the relevance of such term for the collection of documents at hand. It is defined as:

$$e(t) = 1 + \sum_{j=1}^{D} \frac{p(t_j)\log(p(t_j))}{log(D)}$$

where D represents the overall number of documents in the collection, and $p(t_j)$ represents the probability of the term t for the document j. Again, such probability can be approximated by looking at term frequencies. More specifically, if $f(t_j)$ is the frequency of term t in document j, then $p(t_j) = \frac{f(t_j)}{\sum_{j=1}^{D} f(t_j)}$.

In summary, the proposed approach updates PPMI and entropy for all considered terms, and selects those with an entropy value equal to or larger than a pre-fixed threshold WT for each iteration (i.e., for each time window).

3.3 Semantic Representation of Résumés and Job Opportunities

In order to produce semantically relevant representation of résumés and job opportunities we explore the use of the *paragraph vector algorithm* proposed in [11]. This algorithm, also known as *doc2vec*, builds upon the notion of word embeddings by first learning word embedding representations, and then using them in order to generate a vector for the overall document.

As for *word2vec* [12,13], two architectures are available for computing the paragraph vector, namely *Distributed Memory* (PV-DM) and *Distributed Bag-of-Words* (PV-DBOW). The main difference is that, while PV-DM uses a concatenation of paragraph vectors and word vectors for predicting the next word in a window, PV-DBOW simply learns vectors by means of predicting words randomly sampled from the output paragraph.

Our goal is to identify semantic similarity among résumés and job opportunities. Given their semi-structured format, résumés and job opportunities are similar to each other, and use similar words. Therefore, an unsupervised model that considers word ordering such as PV-DM may be easily fooled in assigning a higher similarity score to different profiles due to very similar lexical and syntactic structures. Thus, we use the PV-DBOW for our goal. More specifically, we learn representations of résumés and job opportunities by exploiting only the keywords extracted by the Keyword Extraction algorithm. In particular, we adopt a join of all the keywords extracted by considering the different windows. The advantage is twofold. First, considering only relevant keywords allows learning a higher quality representation of texts. Second, as relevance of keywords is recomputed for each time window, this could ensure a greater ability to discriminate between different skills represented in résumés and job opportunities.

4 Preliminary Evaluation

We have performed several preliminary and exploratory experiments in order to assess the effectiveness of our proposed approach. In this section, we describe the available data, and illustrate results of such experiments.

We performed our experiments on a sample of the whole dataset. More specifically, we considered available résumés for 2016 and 2017, for a total of 12957 résumés. The data were anonymised and provided by IT Partner Italia following the General Data Protection Regulation (GDPR).

For each résumé, several additional information is provided, such as city of provenance and skills of the candidates (i.e. main skill, secondary skill, etc.), as well as manually inserted information concerning an actual evaluation of the candidate from a recruiter. On the one hand, we plan to exploit only information contained in the résumé to build our distributional model. This avoids the necessity of manual labour from recruiters. On the other hand, we exploit skill labels of résumés in order to assess the quality of our representation. Skills are extracted through proprietary algorithms by our industrial partner, and albeit not manually annotated, can provide a good approximation of the skills of a given candidate. We also observe that, due to different sources for résumés in our dataset (e.g. PDF, online forms), conversion into plain text may result in errors and inconsistencies.

We adopted the following parameters. For the PoS patterns, we included nouns, proper nouns and adjectives, and all combinations of three nouns, proper nouns, adjectives, adpositions and determiners, that form multiword patterns in Italian, e.g. "dispositivi Android" (Android devices), "reparto tecnico" (technical department), "sviluppatore Java" (Java developer). For the time window, we considered 180 days. We believe that using a too small time window would yield results that are too fine grained for our goal. Conversely, a window too large could hinder our ability to discover time-specific keywords. Concerning the entropy threshold, we performed several runs and finally selected a value of $WT = 0.7$. In fact, we found that a smaller threshold would yield noisy results, containing many uninteresting n-grams. Conversely, with a higher threshold the algorithm would extract a too small number of n-grams.

Second, we used the paragraph vector algorithm [11] to learn the representation of our data. More specifically, we used the Gensim implementation of the algorithm, known as Doc2Vec[4]. Doc2Vec allows fine tuning of the parameters.

[4] https://radimrehurek.com/gensim/models/doc2vec.html.

We performed several experiments in order to find the best parameters. In particular, we selected the PV-DBOW model. We trained it for 5 epochs, considering a minimum number of occurrences per word equal to 5. The size of the resulting vectors for representation was set to 200. All the other settings were left to default. In addition, we also considered a representation based on *pretrained word embeddings*. More specifically, we used FastText [7] 300-dimensional embeddings for Italian. Each résumé representation is computed as the average of embeddings for words in the text.

It is quite difficult to evaluate our approach in absence of a gold standard or labelled dataset. Thus, we manually selected two sets of résumés, based on their main skill, for an early evaluation. We trust that the skill can represent a reasonable approximation of a résumé. However, we also analysed the actual text in order to make sure we selected suitable candidates. Due to privacy concerns, no additional information over such data can be provided at this time.

In the first set, we selected similar résumés that have the same main skill "Java Junior". On the contrary, for the second set we selected résumés based on very different skill sets. In particular, we considered "Java Senior", "Help Desk intermediate", "Network engineer junior", "Web designer senior", and "Accounting employee". As it is common in many vector space models approaches, we exploit the *cosine distance* to determine similarity among learned representations (i.e. vectors). Results for the two sets and for the two representations (Doc2Vec and pretrained word embeddings) are reported in Table 1.

The analysis of the results shows that our approach is promising. We recall that the similarity is computed only by analysing the text of the résumés and not exploiting the skills. In fact, two different trends emerge for similar and dissimilar documents. More specifically, we observe that résumés for different "Java Junior" figures tend to have a high cosine similarity, while résumés for different professional figures obtain very low cosine similarity among them. On the contrary, when using the pretrained word embeddings representation, cosine similarity is high and very close to each other, thus highlighting that this representation is not very suitable for discriminating different profiles. This may be due to the fact that, whereas word embeddings are trained on general purpose corpora (i.e. Wikipedia), the Doc2Vec representation is learned directly from our domain-specific data. Moreover, in order to validate our approach, we manually evaluated our results, both for keyword extraction and similarity among résumés, with the help of experienced recruiters from the partner company. It appears that our approach can yield good results. However, it is clear that we presented a simple and tentative evaluation of our method, and further analysis is definitely needed. Nonetheless, the underlying idea appears to be promising.

Finally, with the aim of verifying whether the approach described so far is able to discriminate among curricula with different competences and skills, we applied a complete-linkage hierarchical clustering algorithm. We executed the clustering algorithm using both the Doc2Vec and the pretrained embeddings representations. Figures 3 and 4 show the two dendrograms obtained by the two representations, respectively. We can appreciate how actually different clusters

Table 1. Cosine similarities between *Java Junior* profiles and between different profiles using Doc2Vec (a–b) and pretrained word embeddings (c–d).

(a)

ID	1	2	3	4	5
1	1	#	#	#	#
2	0,65	1	#	#	#
3	0,66	0,96	1	#	#
4	0,70	0,95	0,94	1	#
5	0,51	0,53	0,58	0,61	1

(b)

Main Skill	Java Sr.	Help Desk	Network Eng jr.	Web Design sr.	Accounting
Java Sr.	1	#	#	#	#
Help Desk	-0,01	1	#	#	#
Network Eng jr.	0,35	-0,01	1	#	#
Web Design sr.	0,43	-0,10	0,50	1	#
Account. Employee	0,60	-0,08	0,27	0,52	1

(c)

ID	1	2	3	4	5
1	1	#	#	#	#
2	0,77	1	#	#	#
3	0,91	0,78	1	#	#
4	0,91	0,83	0,97	1	#
5	0,91	0,84	0,96	0,97	1

(d)

Main Skill	Java Sr.	Help Desk	Network Eng jr.	Web Design sr.	Accounting
Java Sr.	1	#	#	#	#
Help Desk	0.96	1	#	#	#
Network Eng jr.	0.96	0,96	1	#	#
Web Design sr.	0,93	0,91	0,91	1	#
Account. Employee	0,93	0,97	0,94	0,91	1

of curricula can be identified. Due to space limits, we cannot discuss in detail these clusters and how they can be characterized in terms of competences and skills. However, it is evident that the approach proposed is able to determine groups of curricula with similar characteristics. We can observe, however, that the partitions generated by using Doc2Vec are more homogeneous than the ones generated by employing pretrained embeddings representation. For instance, if we cut the dendrograms at distance 10, we can observe that for the pretrained embeddings representation several clusters contain less than 10 elements, and two macro clusters contain a number of résumés up to around 3500. Conversely, the clustering produced with Doc2Vec at the same distance is more homogeneous. Résumés and keywords are more evenly distributed across clusters.

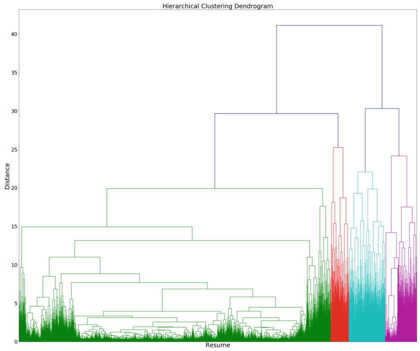

Fig. 3. Doc2Vec representation clustering.

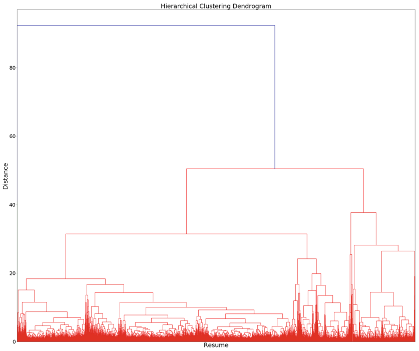

Fig. 4. Pretrained embeddings clustering.

Further, we analysed a number of clusters for the two representations. We realized that the clusters generated by using pretrained word embeddings are generally less relevant to the recruiter. Just to give an example, using Doc2Vec we obtained for example a cluster with frequent keywords such as “web service”, “TCP IP”, “Active Directory”, “MYSQL”, “web application”, that clearly describe the profile of a *web developer*. On the other hand, by analysing clusters obtained with pretrained word embeddings, they appear to be less descriptive. For example, one cluster contains the following frequent keywords: : “C”, “Java”, “Photoshop”, “Google Analytics”, “e-mail”, “marketing”, “assistente amministrativo” (administrative assistant). It is clear that such cluster cannot be considered as a proper representation for a given profile.

5 Conclusions

In this paper, we have presented a system for identifying and extracting time-relevant keywords from résumés and job opportunities, and for building semantically relevant representations of them. Our approach is data-driven and uses semi-supervised strategy for improvement. We have presented our approach and provided a preliminary evaluation of its quality. Although a lot of experimentation is still needed and further improvements are needed, we believe that our direction is promising to automatically extract professional profiles by adopting an unsupervised and data driven approach.

Our final goal is to develop tools that can be implemented in a next-generation ATS, that can take full advantage of information extraction and text mining techniques in order to help recruiters in their day-to-day job.

Acknowledgements. This work was partially supported by Tuscany Region in the context of the project TALENT, “FESR 2014-2020”. We wish to thank Dr. Alessio Ciardini, CEO of IT Partner Italia, and Filippo Giunti, IT manager of IT Partner Italia, for providing us with the data and for the invaluable support and insight, both theoretical and practical, on the recruiting field from a company perspective.

References

1. Bengio, Y., Ducharme, R., Vincent, P., Janvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**, 1137–1155 (2003)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching Word Vectors with Subword Information. arXiv e-prints arXiv:1607.04606
3. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Comput. Linguist. **16**(1), 22–29 (1990)
4. Dumais, S.T.: Enhancing Performance in Latent Semantic Indexing (LSI) Retrieval (1992)
5. Eckhardt, A., Laumer, S., Maier, C., Weitzel, T.: The transformation of people, processes, and IT in e-recruiting: Insights from an eight-year case study of a German media corporation. Empl. Relat. **36**(4), 415–431 (2015)
6. Evert, S.: Corpora and collocations. In: Lüdeling, S., Kytö, M. (eds.) Corpus Linguistics. An International Handbook, article 58, pp. 1212–1248 (2008)
7. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning word vectors for 157 languages. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018) (2018)
8. Harris, Z.S.: Distributional structure. Word **10**, 146–62 (1954)
9. Heggo, I.A., Abdelbaki, N.: Hybrid information filtering engine for personalized job recommender system. In: Hassanien, A.E., Tolba, M.F., Elhoseny, M., Mostafa, M. (eds.) AMLTA 2018. AISC, vol. 723, pp. 553–563. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74690-6_54
10. Laumer, S., Maier, C., Eckhardt, A.: The impact of business process management and applicant tracking systems on recruiting process performance: an empirical study. J. Bus. Econ. **85**, 421–453 (2014)
11. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on Machine Learning (ICML 2014), vol. 32 (2014)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: ICLR Workshop (2013)
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS 2013), vol. 2 (2013)
14. Peters, M.E., et al.: Deep contextualized word representations. arXiv e-prints arXiv:1802.05365
15. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–423 (1948)
16. Shehu, V., Besimi, A.: Improving employee recruitment through data mining. In: Rocha, Á., Adeli, H., Reis, L.P., Costanzo, S. (eds.) WorldCIST 2018. AISC, vol. 745, pp. 194–202. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-77703-0_19