Expert Systems With Applications 123 (2019) 91–107

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Ranking résumés automatically using only résumés: A method free of job offers

Luis Adrián Cabrera-Diego[a,c,1,*], Marc El-Béze[a], Juan-Manuel Torres-Moreno[a,b], Barthélémy Durette[c]

[a] LIA, Avignon Université, 91022 Chemin des Meinajariès, Avignon 84022, France
[b] Polytechnique Montréal, Canada
[c] Adoc Talent Management, 21 Rue du Faubourg Saint-Antoine, Paris 75011, France

ARTICLE INFO

Article history:
Received 18 May 2018
Revised 30 November 2018
Accepted 29 December 2018
Available online 31 December 2018

Keywords:
Résumé
Curriculum vitae
Recommendation system
Relevance feedback
e-Recruitment
Ranking
Mean average precision

ABSTRACT

With the success of the electronic recruitment, now it is easier to find a job offer and apply for it. However, due to this same success, nowadays, human resource managers tend to receive high volumes of applications for each job offer. These applications turn into large quantities of documents, known as résumés or curricula vitae, that need to be processed quickly and correctly. To reduce the time necessary to process the résumés, human resource managers have been working with the scientific community to create systems that automate their ranking. Until today, most of these systems are based on the comparison of job offers and résumés. Nevertheless, this comparison is impossible to do in data sets where job offers are no longer available, as it happens in this work. We present two methods to rank résumés that do not use job offers or any semantic resource, unlike existing state-of-the-art systems. The methods are based on what we call *Inter-Résumé Proximity*, which is the lexical similarity between only résumés sent by candidates in response to the same job offer. Besides, we propose the use of Relevance Feedback, at general and lexical levels to improve the ranking of résumés. Relevance Feedback is applied using techniques based on similarity coefficients and vocabulary scoring. All the methods have been tested on a large corpus of 171 real selection processes, which correspond to more than 14,000 résumés. The developed methods can rank correctly, in average, 93% of the résumés sent to each job posting. The outcomes presented here show that it is not necessary to use job offers or semantic resources to provide high quality results. Furthermore, we observed that résumés have particular characteristics that as ensemble, work as a facial composite and provide more information about the job posting than the job offer. This certainly will change how systems analyze and rank résumés.

1. Introduction

For at least 15 years, the process of attracting possible candidates for a job, i.e., recruitment process, moved from traditional means, like newspapers and job boards, to the Internet and started to be known as *electronic recruitment* or *e-Recruitment* (Kessler, Béchet, Roche, Torres-Moreno, & El-Bèze, 2012; Radevski & Trichet, 2006).

The success of *e-Rectruitment* over traditional recruitment processes lies in the advantages it brings to users and especially to *Human Resources Managers (HRMs)*. Today, due to e-Recruitment, job offers can more easily reach not only specialized communities (Arthur, 2001, page 126) but also wider audiences locally, nationally or internationally (Montuschi, Gatteschi, Lamberti, Sanna, & Demartini, 2014). HRMs' operational costs have been reduced, in certain cases to one-twentieth of the original expenses (Chapman & Webster, 2003). Now, job seekers can search for job offers through the Internet (Looser, Ma, & Schewe, 2013) and apply to them faster by sending an e-mail or filling out a web form with an electronic résumé or CV attached (Elkington, 2005). The greatest e-Recruitment's advantage is the possibility of being in contact with job seekers, employers and HRM all the time around the world (Barber, 2006, page 1).

Although e-Recruitment has helped HRMs with the task of identifying and attracting potential candidates, its use has brought a number of undesirable consequences, especially when high vol-

* Corresponding author.
E-mail addresses: diegol@edgehill.ac.uk (L.A. Cabrera-Diego), marc.elbeze@univ-avignon.fr (M. El-Béze), juan-manuel.torres@univ-avignon.fr (J.-M. Torres-Moreno), durette@adoc-tm.com (B. Durette).
[1] Present address: Department of Computing, Edge Hill University, St. Helens Road, L39 4QP Ormskirk, UK

https://doi.org/10.1016/j.eswa.2018.12.054
0957-4174/

umes of applications are received (Barber, 2006, page 11). After the recruitment process, an HRM must select the group of applicants that are relevant for the job offered. This selection is performed by manually screening résumés.[2] The manual screening consists of examining and comparing applicant information, found in the résumé, with respect to the specifications of the position or *person specification*[3] (Armstrong and Taylor, 2014, Page 226). However, given the large number of applications, HRMs have trouble screening them correctly and rapidly (Trichet, Bourse, Leclère, & Morin, 2004). Furthermore, HRMs have seen an increase in applications from unqualified candidates (Faliagka, Kozanidis, Stamou, Tsakalidis, & Tzimas, 2011), meaning they lose valuable time during the screening process.

The scientific community has proposed multiple systems to reduce the negative impacts of e-Recruitment. The vast majority of the developed systems are based on comparing résumés and job offers, e.g., using measures like Cosine Similarity (Kessler, Béchet, Torres-Moreno, Roche, & El-Bèze, 2009; Singh, Rose, Visweswariah, Chenthamarakshan, & Kambhatla, 2010). In some cases, to improve the matching, they include ontologies or semantic resources that are expected to ameliorate the similarity between documents, like those shown in Senthil Kumaran and Sankar (2013) and Montuschi et al. (2014).

The work that is here presented occurs in the following context. It is the outcome of a collaboration project with a Human Resources enterprise that had a large database of recruitment and selection processes conducted by them previously. The database is divided by job postings[4] in which we can find the applications sent by the interested or directly contacted candidates. Each application is composed, at least, of a résumé and the outcome of the selection process. This database, however, has a particular characteristic, for most of the job postings, neither the job offer nor the person specification is available.[5] This characteristic is due to the software used to store automatically the incoming applications did not provide the option to keep these documents.

Due to the fact that it is impossible to apply state-of-the-art's methods for all the database, we decided to explore how to rank résumés without making use of job offers. The result of this exploration are innovative and simple methods that use uniquely the proximity between résumés sent for the same job posting. To this end, we use a similarity measure and *Relevance Feedback* (Rocchio, 1971) applied with methods based on a similarity quotient and a vocabulary scoring.

Despite in this work, we do not make use of more complex methods, like deep-learning neural networks, or dense text representations, i.e., word embedding, the idea of using them was always present. There were several reasons why not to use these techniques, but the main was that in Cabrera-Diego, Durette, Lafon, Torres-Moreno, and El-Bèze (2015) we started to observe that résumés could be used to rank themselves using similarity measures. Thus, a simple method, like the one here presented could work. Moreover, by using methods based on neural networks, we reduce the chances of understanding and providing the reasons of why a candidate has been chosen to be interviewed, something that it is being looked for, like in Martinez-Gil, Paoletti, and Schewe (2016).

The results obtained from applying our methods over a large set of real recruitment and selection processes, show that our methods, despite not using job offers or semantic resources, can reach great performance. By just applying the method based on résumés proximity, we can rank correctly in average 61% of the résumés sent for a job posting. Nonetheless, this value can reach 93% when it is used along with our proposed Relevance Feedback methods, in which an HRM just need to analyze 20 résumés per job posting, i.e., no more than 50% of the applications sent to the job posting.

In summary, this work present multiple and diverse contributions. The first contribution is that we offer an innovative method, completely different to the ones found in the state-of-the-art, that can rank résumés correctly and automatically. Although this system is used in a very specific context, where job offers are not always present, it can be applied in any condition where the goal is to rank résumés sent to the very same job posting. The second contribution is the use of two different Relevance Feedback that can improve to a great extent other résumé ranking systems. The third and final contribution is the methodology used in this article, which can be used by people to do *a posteriori* analyses of selection processes. For example, HRMs can use the methodology to understand how the selection of candidates was done and which were the keywords that represented the selected and rejected candidates. As well, HRMs can use the tool to determine whether a candidate that should have been called for an interview was left aside. Whereas, psychologist can use the outcome of our methods as a way to determine whether HRM infers aspects like personality (Cole, Feild, Giles, & Harris, 2009) or whether they are affected by errors like misspellings (Martin-Lacroux, 2017). In addition, other systems could use our methods' outputs to generate feedback that rejected candidates could find useful to improve their profiles.

This work is divided into eight sections. In Section 2, we introduce the state-of-the-art methods and our previous work. The methodology and the data are explained in Section 3 and Section 4, respectively. We introduce the experimental and evaluative settings in Section 5. The outcomes from the experiments are presented in Section 6. We discuss the results in Section 7. The work's conclusions and possible future work are presented in Section 8.

2. Related work

In 2002, Harzallah, Leclère, and Trichet (2002) presented the project *CommOnCV* that consists of an automatic analysis and matching of competencies between résumés and job offers. To the best of our knowledge, this was the first project where the scientific community became interested in the automated analysis of résumés. Since the publication of this project, several systems were developed with different approaches and goals. We have grouped the systems into three types: *Résumé matchers, Résumé classifiers* and *Résumé rankers*.

Résumé matchers are systems created for on-line job boards that match uploaded résumés with a job offer or a query, e.g., García-Sánchez, Martínez-Béjar, Contreras, Fernández-Breis, and Castellanos-Nieves (2006); Guo, Alamudun, and Hammond (2016); Radevski and Trichet (2006); Sen, Das, Ghosh, and Ghosh (2012). To achieve the matching of résumés, these systems use mainly on-

[2] According to Thompson (2000), a résumé, also known as *resume, curriculum vitae* or *CV*, is a document prepared by a job candidate, for potential employers, that describes one's education, qualifications and professional experience. In this paper we will use résumé as common term.

[3] This is a document detailing which characteristics, mandatory and optional, should be found in a résumé according to the employer. This document can evolve through the time depending on the job market.

[4] A job posting is composed of three elements: a job offer, a person specification and a set of applications. The job offer is the document that describes the job position (e.g., technician, researcher) but also which are the characteristics that are searched; this document is visible to the job seekers. The person specification, see Footnote 3, is a document only accessible to the HRM and the employer. The set of applications corresponds to the résumés and other documents that are proportioned by the job seekers interested in the job offer.

[5] At the beginning of this work, none of the job postings was linked with its respective job offer. However, after a manual search, we arrived to manually link a portion of job postings, from the database, with their respective job offers. With this subset we created a baseline.

tologies and rules, but they can use some kind of *Relevance Feedback*,[6] like in Hutterer (2011) to improve the match results.

Résumé classifiers consist of systems that bypass HRMs by automatically classifying résumés into relevant or irrelevant candidates. These kinds of systems, such as Kessler, Torres-Moreno, and El-Bèze (2008b) and Faliagka et al. (2013), use machine learning methods to perform this task. In other words, they create a model using data from previous selection processes. The model contains, in theory, the features that make an applicant to appear relevant or irrelevant to an HRM.

Résumé rankers are systems that sort résumés based on proximity between a résumé and a job offer, or even others résumés. As these systems propose rankings, an HRM can decide the point in which résumés become irrelevant for a job and stop reading them. In this kind of systems, proximity between elements can be lexical (Cabrera-Diego, 2015; Kessler, Béchet, Roche, El-Bèze, & Torres-Moreno, 2008a; Singh et al., 2010), semantic (Kmail, Maree, & Belkhatir, 2015; Montuschi et al., 2014; Tinelli, Colucci, Donini, Di Sciascio, & Giannini, 2017) or ontological (Senthil Kumaran & Sankar, 2013). In the following paragraphs we discuss the most representative résumé rankers found in the literature.

E-Gen (Kessler et al., 2009) is a system that can create résumé rankings based on the lexical proximity between résumés and a particular job offer. More specifically, E-Gen compares résumés and a specific job offer using measures such as *Cosine Similarity* and *Minkowski Distance*. The résumés are ranked according to how proximal they are to the job offer. The documents, i.e., résumés and job offers, are represented using a *Vector Space Model*. As well, they make use of a Relevance Feedback method that consists of enriching the job offer vocabulary by concatenating already analyzed relevant résumés from the same job posting. In Kessler et al. (2012) the authors improved the system's performance by adding an automatic text summarization tool to obtain the most relevant information from job offers and résumés.

PROSPECT is a system developed by Singh et al. (2010) that has a résumé ranker among its tools. PROSPECT extracts relevant information from résumés and job offers using *Conditional Random Fields (CRF)*, a lexicon, a named-entity recognizer and a data normalizer. Then, to rank the résumés based on the job offer, PROSPECT compares the information from both documents using *Okapi BM25, Kullback-Leibler Divergence* or *Lucene Scoring*.

We note in the literature the *LO-MATCH* platform (Montuschi et al., 2014). It is a web-based system developed to match professional competencies from résumés and job offers. The LO-MATCH platform is based on ontologies which are used to enhance information from résumés and job offers. The ranking of résumés with respect to a job offer is determined through semantic similarity. LO-MATCH establishes to what degree the words found in a résumé have similar or related meanings to the words occurring in a job offer. The résumés most similar to the job offer are ranked near the top.

EXPERT (Senthil Kumaran & Sankar, 2013) is another system that ranks résumés. However, each résumé and job offer is individually represented by an ontology. To generate each ontology, EXPERT analyzes the information with an ontology and a set of previously defined rules (Senthil Kumaran & Sankar, 2012). EXPERT ranks the résumés by determining how close the job offer ontology is with respect to each résumé ontology. The résumés with ontologies most similar to those of the job offer are ranked near the top.

MatchingSem (Kmail et al., 2015) is a ranking system designed to use multiple ontologies to find the most similar résumés for a job posting. The reason to design a system capable to extract information from multiple ontologies is to represent several domains and/or decrease their lack of coverage. Thanks to ontologies, MatchingSem can create semantic networks that are matched using the *Jaro-Winkler distance*.

I.M.P.A.K.T. (Tinelli et al., 2017) is a platform that allows HRM ranking candidates automatically and obtain the reasons of putting a résumé in a certain position. It is based on *Relational database Management Systems* which help in the creation of improved knowledge bases. As well, the platform allows defining which competencies are required and which are only desired. I.M.P.A.K.T. offers to HRMs information about conflicts or underspecified features found in a résumé.

Another résumé ranker is the one detailed in our previous work (Cabrera-Diego, 2015). There, we present the first version of the method that in this work is extended and improved. It consists of using a measure that we call *Average Inter-Résumé Proximity (AIRP)*. This measure determines the relevance of a résumé according to how similar it is to other résumés from the same job posting. To improve the ranking of résumés, we use Relevance Feedback and apply it with a factor that increases when a résumé is closer to those considered by an HRM as relevant.

In the last years, some other researchers have worked on tasks related to the automatic ranking of résumés. For example, in Martinez-Gil et al. (2016) the authors propose an approach to improve the ranking of résumés by *matching learning*; as well, how to use matching learning to represent, in the future, documents using a common vocabulary. As well, related to the previous work, Martinez-Gil, Paoletti, Rácz, Sali, and Schewe (2018) propose a theory of how to match résumés and job offers, but also ranking them by using knowledge bases, lattice graphs and lattice filters. Another example is the analysis of social media to evaluate the emotional intelligence of candidates (Menon & Rahulnath, 2016). In Zaroor, Maree, and Sabha (2017), for instance, résumés and job offers are classified automatically in occupational categories; semantic networks are used to find the best matching between these documents.

3. Methodology

Our methodology is composed of two parts. In the first part, we determine the similarity of résumés in order to rank them. In the second, which is optional although suggested, we ask the HRM for Relevance Feedback and apply it. More specifically, the methodology used in this article is composed of five steps which are graphically represented in Fig. 1.

In step *I*, we calculate the proximity between pairs of résumés using *Inter-Résumé Proximity* (Section 3.1). Once all the proximity values have been calculated, we estimate the Average or Median Inter-Résumé Proximity for each résumé in step *II* (Section 3.2). It is in this step where we formulate the hypothesis that the resulting values indicate the relevance of the résumés for the job posting. In step *III*, we sort the scores obtained in step *II* in descending order to rank the résumés.

If we want to improve a ranking, we can make use of Relevance Feedback (Section 3.3). This process starts in step *IV* where an HRM analyzes a small set of résumés in order to determine whether they are relevant or not. Furthermore, they can identify and sort the terms that represent better relevancy. Once the HRM has finished, the Relevance Feedback is processed in step V. In this case, we can process the Relevance Feedback using the Relevance Factor (Section 3.3.1) and Vocabulary Scoring (Section 3.3.2). The output of the Relevance Feedback is then introduced in step *III* to re-rank the remaining résumés, i.e those not seen during the Relevance Feedback.

[6] Relevance Feedback is the interaction of a human user with an information retrieval system, in order to evaluate its results and to modify requests for improving data retrieval Rocchio (1971).

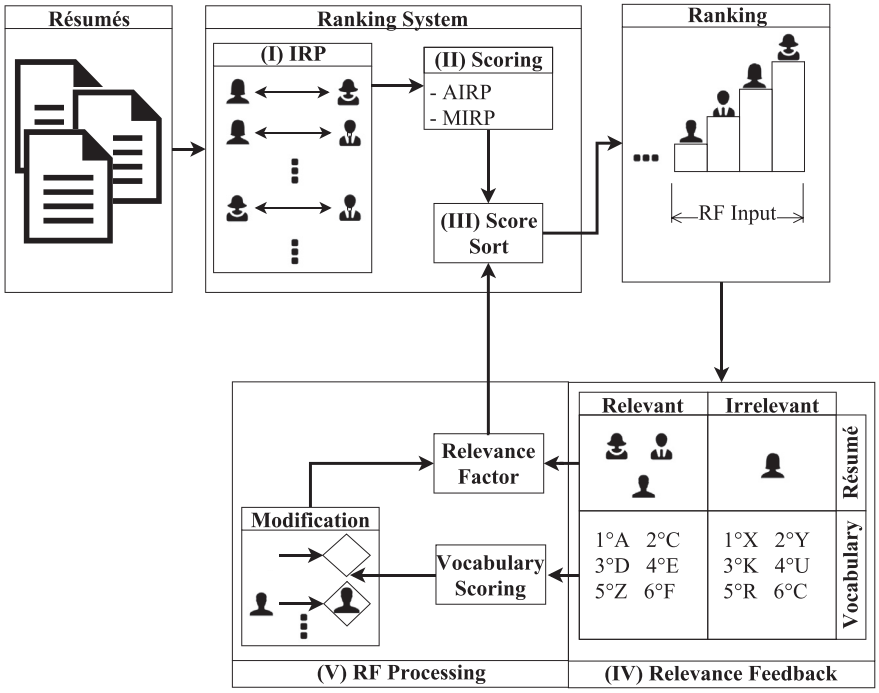

Fig. 1. Methodology overview.

3.1. Inter-Résumé Proximity

The *Inter-Résumé Proximity (IRP)* is defined as the degree of similarity between two résumés that were sent by different candidates applying for the same job positing. To mathematically define the IRP, consider J as the set of résumés gathering all the candidates that applied to the same job posting, $J = \{r_1, r_2, r_3, \ldots r_j\}$. Every résumé r in J is unique and from a different applicant, i.e. there are no duplicated résumés or candidates in the job posting. We present the definition of Inter-Résumé Proximity (IRP) by Eq. (1).

$$\mathrm{IRP}(r, r_x) = \sigma(r, r_x); \forall r \neq r_x; r, r_x \in J \tag{1}$$

where r and r_x are two different résumés from J; σ is a proximity measure.

In this study, we use *Dice's Coefficient* as σ because in Cabrera-Diego et al. (2015) we observed, through statistical analyses, that this similarity measure is the most adequate for this task.[7] Although Dice's Coefficient is frequently defined in terms of sets, as in Eq. (2), we have redefined it in Eq. (3) to be used in a vector representation.

$$\text{Dice's Coefficient}(r, r_x) = \frac{2 \cdot |r \cap r_x|}{|r| + |r_x|} \tag{2}$$

$$\text{Dice's Coefficient}(r, r_x) = \frac{2 \cdot \sum_i^n min(\alpha_i, \alpha_{xi})}{\sum_i^n \alpha_i + \sum_i^n \alpha_{xi}} \tag{3}$$

where $r = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ and $r_x = \{\alpha_{x1}, \alpha_{x2}, \ldots, \alpha_{xn}\}$ are vector representations of the résumés r and r_x respectively. Each vector has n dimensions and their components are expressed by α; *min* is a function that outputs the smallest component between r and r_x in each vector dimension.

Note that Dice's Coefficient has a closed interval [0, 1], where 1 means that both documents are identical and 0 indicates they are completely different and have nothing in common.

[7] Other measures tested in Cabrera-Diego et al. (2015) were Cosine Similarity, Jaccard's Index, Manhattan distance and Euclidean distance. However, it was Dice's Coefficient the one that presented the best performance in the analysis of résumés.

3.2. Average and median Inter-Résumé Proximity

In Cabrera-Diego et al. (2015) we determined through a statistical analysis that, on average, the similarity between relevant résumés is greater than the similarity between irrelevant ones. Equally, we observed that relevant résumés tend to be dissimilar to the group of irrelevant résumés. From this outcome, we can infer that relevant résumés should have multiple terms in common, while irrelevant résumés should present a variety of terms that are not shared, either by other irrelevant résumés or by the relevant ones. Based on this interpretation, we designed what we call the *Average Inter-Résumé Proximity (AIRP)*. It is a method of finding relevant résumés based on their proximity to other résumés. The concept is that a relevant résumé will have, on average, higher values of IRP than an irrelevant résumé.[8]

The mathematical definition of AIRP is presented by Eq. (4).

$$\mathrm{AIRP}(r) = \frac{1}{j-1} \sum_{x=1}^{j} \mathrm{IRP}(r, r_x) \tag{4}$$

where r is a résumé selected for analysis from J, r_x is another résumé related to J but different from r and j is the number of résumés sent to J.

We introduce as well the *Median Inter-Résumé Proximity (MIRP)*. It is a variation of AIRP, but it consists of calculating the median instead of the average of a set of Inter-Résumé Proximity values. The main reason to use this central-tendency measure is that it is more robust against skewness and outliers[9] than the mean. The formula for calculating the MIRP is given by Eq. (5).

$$\mathrm{MIRP}(r) = \mathrm{MEDIAN}[\mathrm{IRP}(r, r_x)]_{x=1}^{j} \tag{5}$$

[8] A relevant résumé should have high values of IRP with respect to other relevant résumés and low values of IRP with respect to irrelevant ones. However, irrelevant résumés should have constantly low values of IRP in accordance with the analyses done in Cabrera-Diego et al. (2015).

[9] An outlier is a value with an atypical magnitude with respect to the total set (Mason, Gunst, and Hess, 2003, page 70).

where r and r_x are two different résumés from J and j is the number of résumés sent to J.

3.3. Relevance Feedback

In addition to AIRP and MIRP, we propose to use Relevance Feedback as a method for validating and enriching the information used by our ranking methods.

In our study, Relevance Feedback is the process where an HRM determines which résumés, from a sample of the ranking given by AIRP or MIRP, are relevant and irrelevant for the job posting. Furthermore, an HRM can indicate the terms that better characterize the relevant and irrelevant résumés found during the previous step. Based on these inputs, we process and apply the feedback to offer an improved ranking of the remaining résumés. The Relevance Feedback given for one job posting does not affect the way we rank other job postings, as the inputs can differ.

We propose two methods for applying Relevance Feedback. The first method, called Relevance Factor and presented in Section 3.3.1, consists of calculating a quotient that takes into account the Inter-Résumé Proximity between a résumé and those considered relevant or irrelevant during Relevance Feedback. This method, as seen in Fig. 1, is introduced into the ranking process by a simple multiplication during the calculation of either AIRP or MIRP. The second method (Section 3.3.2) resides in weighting the terms indicated by the HRM that better represent the relevant and irrelevant résumés seen during Relevance Feedback. Because of its characteristics, explained in its respective section, this last method modifies the Relevance Factor.

3.3.1. Relevance factor

The first method for introducing Relevance Feedback consists of determining the proximity between the remaining résumés from a job posting and those, from the same job posting, that were analyzed during the Relevance Feedback. We achieve this with a formula that we have called *Relevance Factor* (RFa). The Relevance Factor goal is to improve the ranking of résumés. Thus, on one hand, the Relevance Factor pushes to the ranking's top the résumés that are more proximal to those considered as relevant during the Relevance Feedback. On the other hand, it pulls down, to the ranking's bottom, those résumés which are more proximal to the irrelevant ones.

Let us consider $F = \{r_1, r_2, \ldots, r_f\}$ as the set of résumés sent by applicants for a job posting J that were analyzed during a Relevance Feedback process. Each résumé from F was classified by an HRM into one class, either relevant (R) or irrelevant (I). We have defined the Relevance Factor, RFa, in Eq. (6).

$$\mathrm{RFa}(r) = \frac{\Omega + \sum \mathrm{IRP}(r, r_{xR})}{\Omega + |R|} \cdot \frac{\Omega + |I|}{\Omega + \sum \mathrm{IRP}(r, r_{xI})}; \quad \forall r_{xR} \in R;\ r_{xI} \in I;\ r_{xR}, r_{xI} \in F \tag{6}$$

where r is the résumé to be analyzed, R and I represent the set of résumés considered, respectively, as relevant and irrelevant during the Relevance Feedback process. Furthermore, Ω is a constant, empirically set to 1×10^{-10} which is used to avoid undetermined values[10] and IRP is the function described in Eq. (1).

The behavior of the Relevance Factor depends on the interval of the proximity measure used to determine IRP (Eq. (1)). Since we use Dice's Coefficient, the Relevance Factor will be greater than one ($\mathrm{RFa}(r) > 1$) when the résumé r is more proximal to the relevant résumés. It is going to be $\mathrm{RFa}(r) = 1$ if r is equally similar to relevant and irrelevant résumés. And, if the résumé r has more in common with the irrelevant résumés, the Relevance Factor will approach to zero.

[10] In some cases during the Relevance Feedback, it is possible to find only relevant or irrelevant résumés, but not both. Without this constant one side of the formula would be 0/0.

Table 1
Example of how the Relevance Factor would be calculated for three résumés, *A*, *B* and *C*, that belong to a hypothetical job posting J containing eight different résumés, $J = \{R_1, R_2, R_3, I_1, I_2, A, B, C\}$. The example considers that J has three relevant résumés (R_1, R_2, R_3) and two irrelevant ones (I_1, I_2) previously detected by an HRM during a Relevance Feedback process.

r	r_{xR}	IRP(r, r_{xR})	Σ IRP(r, r_{xR})	r_{xI}	IRP(r, r_{xI})	Σ IRP(r, r_{xI})	RFa(r)
A	R_1	0.90	2.45	I_1	0.20	0.50	$\frac{2.45}{3} \cdot \frac{2}{0.50} = 3.26$
	R_2	0.75		I_2	0.30		
	R_3	0.80					
B	R_1	0.35	1.35	I_1	0.40	0.90	$\frac{1.35}{3} \cdot \frac{2}{0.90} = 1.00$
	R_2	0.55		I_2	0.50		
	R_3	0.45					
C	R_1	0.30	0.90	I_1	0.80	1.55	$\frac{0.90}{3} \cdot \frac{2}{1.55} = 0.38$
	R_2	0.40		I_2	0.75		
	R_3	0.20					

The introduction of the Relevance Factor into the ranking of résumés is done by simple multiplication, i.e., the Relevance Factor of a résumé is multiplied by its respective score determined by either AIRP or MIRP.

To understand the Relevance Factor better, it should be indicated that Eq. (6), can be split into two parts. The left side calculates IRP with respect to the relevant résumés, while the right side is in accordance with the irrelevant résumés. We describe in the following paragraph a hypothetical process of its calculation.

Let us consider a job posting J composed of eight different résumés, $J = \{R_1, R_2, R_3, I_1, I_2, A, B, C\}$. During a Relevance Feedback process, an HRM analyzed five of these résumés, i.e., $F = \{R_1, R_2, R_3, I_1, I_2\}$, and found out that three were relevant (R_1, R_2, R_3), while two were irrelevant (I_1, I_2). In Table 1, we present how the Relevance Factor would be calculated for the résumés that were not analyzed by the HRM (*A, B, C*). As it can be observed in Table 1, the résumé A is very similar to relevant résumés, therefore, its $\mathrm{RFa}(A) = 3.26$; this means that its score, either AIRP or MIRP, will be multiplied by a factor of 3.26. Regarding résumé B, it has a $\mathrm{RFa}(B) = 1.00$, this means that it is equally similar to relevant and irrelevant résumés; the AIRP or MIRP of B will rest the same. Concerning résumé C, it has a $\mathrm{RFa}(C) = 0.38$ due to its high similarity to irrelevant résumés and, in consequence, its AIRP or MIRP will be affected by a factor of 0.38.

3.3.2. Vocabulary Scoring

The second method for applying Relevance Feedback consists of processing the vocabulary that, in accordance with the HRM, better represents the résumés marked as relevant or irrelevant during the Relevance Feedback. The objective is to adjust the weights of the terms that cause a candidate to be considered by an HRM as relevant or irrelevant for the job posting. To achieve this, during the Relevance Feedback an HRM indicates and sorts which terms, seen in the analyzed résumés, characterized what made a candidate to be relevant or irrelevant. The sorting of the terms should be done regarding their representativeness.

Formally, consider $V_c = \{t_1, t_2, \ldots, t_v\}$ as the vocabulary selected and sorted by an HRM that better represents the résumés from class c during Relevance Feedback. For each term from V_c, we compute its Term Score $T_c(t)$, i.e., a value that allows us to boost or minimize the terms that define each class c. In Eq. (7), we define the Term Score $T_c(t)$ for a term t appearing in V_c.

$$T_c(t) = \sqrt[5]{\frac{1}{\mathrm{rank}(t)}}; \forall t \in V_c \tag{7}$$

where $\mathrm{rank}(t)$ is the position of term t defined by an HRM in V_c. The Term Score always has a value within the half-closed interval

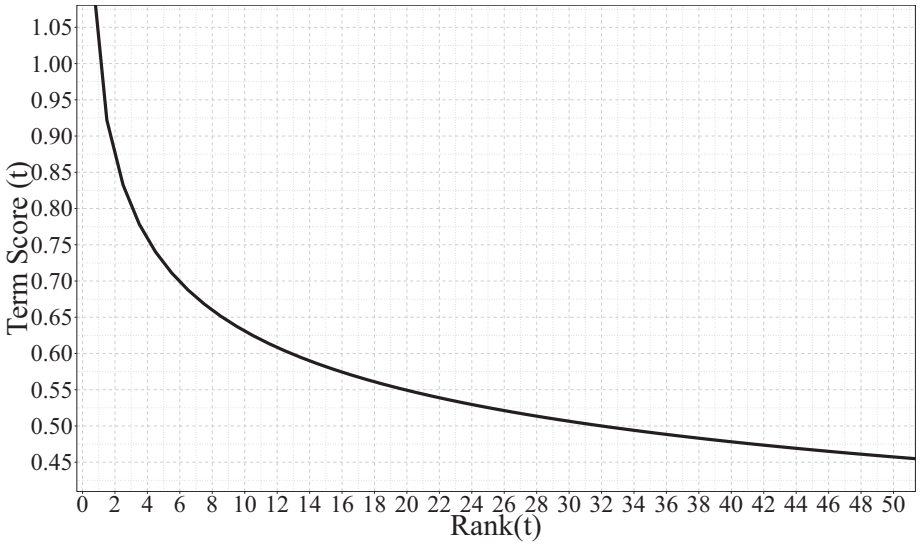

Fig. 2. Plot of the term score for ranks of t between 1 and 50.

[1, 0). A term with a value close to 1 expresses a high representativeness of class c, while a term with a value near to zero means that it hardly represents class c and should be minimized.

Using a root in Eq. (7), specifically the 5^{th} root, should be discussed. We empirically chose this function for two reasons. First, it allows us to create a score between 1 and 0. Second, it slowly decreases and preserves the sense of representativeness provided by the HRM, i.e., the way that terms were sorted by the HRM is kept. It can be seen in Fig. 2 how the Term Score changes in accordance to the rank of t; for instance, the term that is ranked first has a Term score equal to 1; the second ranked term has a score $T_c = 0.870$; and for the fiftieth score, $T_c = 0.457$.

As there is always a set of terms that will not appear in V but that are found in other résumés for the same job posting J, it is essential to give to these terms a Term Score T_c in order to keep the model balanced. In other words, we cannot leave the terms that did not appear in a V_c with higher values than those that were analyzed by an HRM. We assign a value of 0.01 to all the terms belonging to the résumés of J that are not present in a V_c.[11] This figure was chosen empirically to minimize the terms that are not representative of the model without deleting them.[12]

Since the Term Score $T_c(t)$ of every term t is different for each class c (relevant and irrelevant), we use the Term Score uniquely within the Relevance Factor (Section 3.3.1), as it calculates the Inter-Résumé Proximity with respect to relevant and irrelevant résumés separately. To be more specific, the Term Scores only modify terms' weights of each class used at the computation of Inter-Résumé Proximity in Eq. (6).

3.3.3. Selecting the résumé s for the Relevance Feedback

Even though the Relevance Feedback described in Rocchio (1971) consists of choosing a number of top-retrieved documents, we test whether the Relevance Feedback determined with other position-retrieved documents is useful to HRMs. More specifically, we use the Relevance Feedback of the documents retrieved from the following positions:

- Top. This is the classic method which consists of taking the top ranked résumés to improve the following rankings. In the case where we find non-relevant résumés among the top, it may be a way to determine which characteristics, although common, may not be required for the job or are not the ones searched by the HRM.
- Bottom. This is the opposite of the classic method, as we select the résumés located at the end of rankings. We infer that finding a relevant résumé with a low ranking can provide more useful feedback than detecting an irrelevant résumé at the top. Furthermore, this may be interesting for the Human Resources domain, as leaving a relevant résumé at the bottom would set aside the objectives of the rankings.
- Top and Bottom (henceforth Both). For this position, we decided to merge the ideas from the first two Relevance Feedback positions. More specifically, in this position we ask a recruiter whether some résumés from the top and the bottom are truly relevant.[13] The goal is to reduce the weaknesses of the Bottom and Top positions; we may detect truly relevant and irrelevant documents ranked first and also those that are interesting but mis-positioned at the end of a ranking.

In addition to the different Relevance Feedback positions, we decided to test whether an iterative application of Relevance Feedback could improve the résumé rankings more quickly. Under non-iterative conditions, once the Relevance Feedback has produced a new ranking the process ends. Nonetheless, for iterative conditions, once a new ranking is produced, it can be re-analyzed by an HRM in a new Relevance Feedback process.

4. Data

For this article, we used a set of 171 job postings which were processed (recruitment and selection) by a French human resources enterprise between November 2008 and March 2014. These job postings come from different professional domains (e.g., chemistry, communications, physics and biotechnology) and position levels (e.g., laboratory researcher, intern, project manager and engineer). These 171 job postings were chosen because they contain at least 20 unique résumés in French; at least 5 of them are

[11] For instance, a term t can appear in $V_{\text{Irrelevant}}$ but not in V_{Relevant}. Thus, for this same t the Term score V_{Relevant} will be 0.01.

[12] We experimented with other values: 0.25, 0.1 and 0.05. We observed that by decreasing the value the results were improved.

[13] Half of the résumés for the Relevance Feedback are from the top. The other half belong to the ranking's bottom.

relevant, and 5 are irrelevant.[14] In total, the corpus contains 14,144 French résumés divided among these 171 job postings.

All the job postings are composed of applications, and each application contains the documents associated with the recruitment and selection process. It is important to note that not all the documents located inside the applications corresponded to résumés; we could find motivation and recommendation letters, diplomas, interview minutes and social network invitations as well. To obtain only the French résumés, we made use of a résumé detector. The résumé detector is a linear *Support Vector Machine (SVM)* developed previously in Cabrera-Diego et al. (2015). Furthermore, all the résumés were lower cased and lemmatized; for lemmatizing the documents, we used Freeling 3 (Padró & Stanilovsky, 2012). Stop-words, punctuation marks and numbers were deleted. In addition, all duplicated résumés within the same job posting were deleted.[15] See Cabrera-Diego et al. (2015) in order to learn more about this pre-processing task.

According to the HRMs with whom we worked, the system employed to manage the applications allowed them to organize each applicant into one of the following selection phases: *Unread, Analyzed, Contacted, Interviewed* and *Hired*. The phases were assigned to each applicant depending on the last point to which they arrived. For this article, we grouped four of these phases into two different classes: *relevant* and *irrelevant*.

The first class, relevant, corresponds to the phases Contacted, Interviewed and Hired. It represents the applicants who after reading their résumés were approached by a recruiter. The second class, irrelevant, contains only the résumés that remained in the Analyzed phase, i.e., the applicants that were not approached by an HRM after reading their résumés.

With respect to the applications that remained in the Unread phase, these were discarded from the analysis since we cannot infer whether they were relevant or irrelevant for the job. Furthermore, most of these applications were not read because the selection process ended as they were received.

There are two reasons to classify four of five phases into two classes. The first is that to determine whether an applicant will be hired implies the analysis of elements that are not present in a résumé (e.g., interview results, expected salary, job location and withdrawal). The second one is that we do not want to replace humans with an automaton in the selection process. Instead, we want to assist humans during the most difficult part of the selection process, which is in discerning relevant and irrelevant applicants. And this can be achieved by ordering applicant's résumés in terms of how relevant are for the job posting.

It is important to note that some applications from the corpus, although impossible to trace, started as *Direct contact*. This means that an HRM found, usually on the Internet or job seekers databases, the résumé of a person who fulfilled the person specification and decided to contact this person directly. Thus, for some job postings the relevant résumés can accurately reflect the searched profile. This action can affect the number of relevant applicants for a job posting, which in some cases can be equal or greater than the number of irrelevant applicants. However, this characteristic from the corpus should be seen as normal, since for an HRM to make direct contact is a way to speed up the recruitment and selection processes.

Furthermore, it should be indicated that we do not combine applications from different job postings, even if they belong to similar job positions. The reason is that each job posting is linked to a job offered by a specific enterprise, in a particular date and with a set of desired characteristics. In other words, each job posting might attract different job seekers despite describing a very similar job position; aspects like years of experience, spoken languages, mobility, relocation and salary can affect how the job market reacts. This variability makes impossible to determine whether a candidate from one job posting would participate in another one or whether a candidate would be considered equally relevant.

To conclude with this section, after a manual search, we arrived to link 60 of the 171 job postings with their respective job offer. With these 60 job postings we created a baseline that will be described in Section 5.

4.1. Data representation

We decided to represent each résumé from the corpus as a set of *n*-grams in a *Vector Space Model (VSM)* (Salton, Wong, & Yang, 1975). To be specific, for each résumé we extracted its set of unigrams, bigrams and trigrams. Every set of *n*-grams was saved as a vector, one per résumé. The vectors' component weights (W) are the relative frequency of each *n*-gram which could be multiplied by a weight modifier (Δ); we present W in Eq. (8).

$$W(\bullet) = \mathbb{F}(\bullet) \cdot \Delta(\bullet) \tag{8}$$

where $\bullet$ is an *n*-gram and $\mathbb{F}$ is the relative frequency calculated with respect to each résumé. The weight modifier Δ can be one of the following:

- $\Delta = 1$. In this case, we represent the data only by the relative frequency of each *n*-gram.
- $\Delta = \text{IDF}(\bullet)$. Each *n*-gram ($\bullet$) is weighted with respect to a *Term-Frequency Inverse-Document Frequency (TF-IDF)* Spärck-Jones (1972).[16]

Once the résumés of a job posting have been ranked for the first time, either with AIRP or MIRP, and a vocabulary scoring has been set, a new Δ for Eq. (6) can be used:

- $\Delta = T_c(\bullet)$. In this case each *n*-gram ($\bullet$) is modified by its respective Term Score T_c (see Eq. (7)); where c is the class (relevant or irrelevant) that will affect uniquely.
- $\Delta = \text{IDF}(\bullet) \cdot T_c(\bullet)$. It is similar to the previous Δ, however, it can be modified by IDF in the case, the original representation made use of the weight too.

In all the cases, these last two Δ do not affect permanently the weights of the terms, they are only locally used each time Eq. (6) is called.

4.2. Data for the Relevance Feedback

Although the ideal experimentation would consist in applying our methods and asking HRM for Relevance Feedback on real time, the fact is that this task would be very expensive. Moreover, the HRM would have to do this task besides their normal work duties and it would be hard to get accurate results in cases where the person specification evolved over time. Thus, we decided to simulate the Relevance Feedback.

Regarding the Relevance Feedback in which an HRM indicates whether a résumé is relevant or irrelevant, we made use of the information available in the corpus. As we explained at the beginning of Section 4, every application and, therefore, every résumé

[14] All the résumés must be either relevant or irrelevant, but each job cannot have less than 5 per class.

[15] There were job postings in which the same applicant sent their own résumé multiple times. Thus, to avoid a bias, we deleted the duplicated résumés with a set of heuristics developed in Cabrera-Diego et al. (2015). Among the heuristics used, we can highlight the selection of the most recent résumé in the application folder or the detection of the exact same applicant e-mail.

[16] The IDF for each unigram, bigram and trigram was calculated using all the corpus described at the beginning of Section 4 (14,144 résumés).

belongs to a real selection process. Thus, at a given moment, every résumé was analyzed by an HRM who considered whether it was from a relevant or irrelevant applicant. The information found in the corpus allows us to create a simulation that can be reproduced again if necessary.

For the vocabulary scoring, we decided to explore three simulations, S_1, S_2 and S_3, in which we select and weight differently the terms for the vocabulary scoring. Each simulation is composed of 100 n-grams in total, 50 describing the relevant résumés and 50 the irrelevant ones. These simulations are different from the one used to determine whether a résumé is relevant or irrelevant, as the corpus does not contain this kind of information. However, they are based on information found in the corpus and in consequence reproducible. In the following subsection, we explain in detail how S_1, S_2 and S_3 were determined.

4.2.1. Simulations for Vocabulary Scoring

Consider $V = \{t_1, t_2, \ldots, t_v\}$, the vocabulary composed of the n-grams (t) that occur in at least 2 résumés from the Relevance Feedback.[17] The process to generate the three simulations is as follows:

1. For each term t belonging to V, we calculate the squared probability of term t occurring in each possible class c, either relevant or irrelevant. This is done using Eq. (9):

$$p_c^2(t) = \left(\frac{D_c(t)}{D(t)}\right)^2 \tag{9}$$

   where $D_c(t)$ is the number of résumés belonging to class c; $D(t)$ is the number of résumés analyzed in the Relevance Feedback. The equation is an adaptation of Gini's Coefficient[18] presented in Cossu (2015). For a set of classes $\mathbb{C} = \{c_1, c_2, \ldots, c_k\}$, Gini's Coefficient has an interval between $[1/k, 1]$, where $1/k$ means that a term appears in every class, while 1 indicates that the term belongs to one class (Torres-Moreno et al., 2012).
2. Then, we calculate a factor (f_c) that takes into account the number of documents from class c where the n-gram appeared and the sum of the n-gram's weights (W) inside these documents. The factor is presented in Eq. (10).

$$f_c(t) = D_c(t) \cdot \sum W_c(t) \tag{10}$$

   where t represents an n-gram, c is one of the two possible classes (relevant or irrelevant), f_c is the factor for the class c, D_c is the number of documents of class c where t appears and W is the n-gram weight (see Eq. (8)).
3. For each class c we sort the n-grams first according to their squared probabilities p_c^2 and then by their factor f_c. If two or more n-grams share the same squared probabilities and factors, although this is unusual, we assign them different but consecutive locations in the sorted list.
4. We select the first 50 n-grams for each class, to which we calculate their Term Scores (T_c) using Eq. (7).
5. For the rest of n-grams, or those that did not occur in the résumés from the Relevance Feedback, we give them a Term Score of 0.01 as explained in Section 3.3.2.

Simulation S_1 consists of selecting and scoring the vocabulary according to the information found only in the résumés used for Relevance Feedback. In other words, the résumés from the Relevance Feedback are used to calculate the squared probabilities and the factors of the n-grams. Next, for each class c, we calculate the Term Scores for the first sorted 50 n-grams.

For simulation S_2, we decided to recreate a scenario where the selection and sorting of the terms is done carelessly. Put differently, the terms that, in theory, represent relevant and irrelevant résumés are ignored and are not used in the Relevance Factor (Eq. (6)). To this end, we sort the n-grams using only the information from the Relevance Feedback, as we do for S_1, but the Term Score of the first 50 n-gram of each class c is set to zero ($T_c = 0$).[19] For the rest of terms, the Term Score is the default one, i.e. 0.01.

In simulation S_3, we try to model optimally the n-grams that would be chosen by an HRM in real life. To this end, we calculate the squared probabilities and factors f_c based on the information in all the résumés from the job posting. However, we continue to sort and calculate the Term Scores for the terms that only occur in the résumés from Relevance Feedback. In summary, we can have high reliable squared probabilities and factors f_c but we only affect the n-grams that would have been seen by an HRM during the Relevance Feedback.[20]

5. Experimental and evaluative settings

There are multiple experiments that can be done following different configurations, however, although we explored a large amount of possible combinations, due to space limitations we only present the experiments that could contribute the most to the state-of-the-art. The experiments realized are summarized in the following list:

- *No Relevance Feedback:* We apply our methods without using any kind of relevance feedback, and we compare them against a couple of baselines.
- Relevance Feedback applied using
  - *The Relevance Factor:* We explore how different Relevance Feedback position (Top, Bottom and Both) affect the Relevance Factor. As well, we analyze whether the iterative application of the Relevance Factor can improve faster the ranking of résumés.
  - *The Relevance Factor with Vocabulary Scoring:* We analyze how the simulations of Vocabulary Scoring affect the rankings created by the Relevance Factor.

Two different baselines are used, the first one consists of a system that generates a random ranking for each job posting. The second baseline resides in using the 60 job postings to which we arrived to link with their respective job offer and calculate the similarity résumés/job offer. More specifically, for each job posting, we apply Dice's Coefficient between its job offer and every element from its set of résumés. Job offers are pre-processed under the same parameters that the résumés, as explained in Section 4. Although a comparison with other methods or systems from the state-of-the-art would have been desired, to the extent of our

[17] Because we simulate the vocabulary scoring, to use terms that were seen only in one résumé may not be reliable but speculative. In fact, a one time-seen term, and in consequence its pertinence, may be no more than a coincidence which could change by increasing the number of documents analyzed.

[18] Although Gini's Coefficient is frequently used in economics for wealth distribution, it has been used in other NLP works, e.g., Fang and Zhan (2015) and Cossu, Janod, Ferreira, Gaillard, and El-Bèze (2014). Gini's Coefficient in NLP has the objective of modifying the weight of an element in the data model by determining to which degree it represents a certain class or set of them (Torres-Moreno, El-Bèze, Bellot, & Béchet, 2012).

[19] By making zero the Term Score of these n-grams, we affect their weight in the vector space model as explained in Section 4.1. This modification has, in consequence, an effect in the Relevance Factor (Eq. (6)), where the résumés containing most of the terms representing a class, instead of being pushed up or pulled down, they will stay in the same position in the rank.

[20] In simulation S_3 is possible that after sorting the n-grams, the one placed in the first place does not appear in the Relevance Feedback. Thus, as this n-gram could not have been seen by the HRM during the Relevance Feedback, we must consider another n-gram as the one in the first place. This will be the first term seen in the Relevance Feedback that has the best squared probability and factor f_c. For the following terms the rules are the same.

knowledge, none of the systems or datasets have been released to the public.[21]

In the case of the experiments with Relevance Feedback, we have restricted the feedback size to a range between 2 and 20 résumés. It should be noted that we never use more than 50% of the résumés for each job as feedback. In fact, the 171 job postings described in Section 4 were chosen because they had at least 20 résumés, from which at least 5 were from relevant applicants and 5 from irrelevant ones. When we use more than 10 résumés for the Relevance Feedback, we always verify that there is at least twice the résumés for the job posting, with more than 1/4 of them being relevant and no less than 1/4 irrelevant. For example, to do a Relevance Feedback of 16 résumés, a job posting must have at least 32 résumés in total, and no less than 8 must be relevant or irrelevant. If one job posting do not have these characteristics then it is discarded, for that size of Relevant Feedback, from the analysis. All these precautions are taken to avoid inflating the measurements for evaluation artificially.

In the experiments related to the iterative application of Relevance Factor, we explore how rankings are affected when multiple and sequential Relevance Factor processes are done. In other words, we start by doing a Relevance Factor over 2 résumés. This process will rank the remaining résumés of the job posting in an improved way. After that, a new process of Relevance Feedback is done on which 2 new résumés are analyzed. The Relevance Factor is calculated again and the process is repeated until having revealed up to 20 résumés.

It should be mentioned that the corpus had 171 job postings that fulfilled the characteristics used for the Relevance Feedback up to size 10. For a Relevance Feedback of size 20, there were only 127 job postings with the established characteristics.

All the calculations for AIRP and MIRP were parallelized using GNU Parallel (Tange, 2011), a shell tool created to run the same task multiple times but with different inputs. More specifically, the parallelization consists in assigning a CPU thread to each job posting. Therefore, multiple job postings can be run at the same time.

We decided to evaluate each ranking of résumés using *Average Precision* (AP) (Buckley & Voorhees, 2000). AP is an evaluation metric designed for rankings with two grades of relevance: relevant and irrelevant.[22] Furthermore, AP determines, at the same time, the precision and the recall of a ranking in accordance to the position of its elements (Voorhees & Harman, 2001). In order to have a good value of AP, i.e., close to 1, the relevant elements should be positioned at the top of a ranking, while those that are irrelevant should be located at the bottom of a ranking. In our case, a ranked résumé is considered to have the correct relevance when it is similarly marked in the corpus data (see Section 4).

To evaluate the performance of the methods used to rank résumés, we calculate the *Mean Average Precision* (MAP) for each one (Buckley & Voorhees, 2000). As the name indicates, the MAP consists of averaging all the AP values obtained using the same method.

In order to verify whether the MAP values obtained for each tested method are significantly different, we analyze the results using a one-way *Repeated Measures Analysis of Variance* (rANOVA).

The assumptions of rANOVA, data normality and sphericity, are tested with the *Shapiro-Wilk Test* and the *Mauchly's Test*, respectively. In both cases, the alpha to refute the null hypothesis is set to 0.05.

The results from the rANOVA are considered to be significantly different when the p value is less than 0.05. In the case we compare more than two methods, and the rANOVA show a significant difference, we also make use of a *post hoc* test. More specifically, we utilize a *Pairwise t-Test* with $\alpha = 0.05$ in order to determine which pairs of groups are significantly different.

For each pair of experiments showing a significant difference, we calculated the effect size using *Cohen's d*. Effect sizes are values that helps to quantify the difference between two analyzed groups. As thumb rule, effect size can be classified into small ($d = 0.2$), medium ($d = 0.5$) and large ($d = 0.8$) (Cohen, 1988, Page 20).

The statistical analyses were performed using *R* (R Core Team, 2018).

Table 2
Summary of the statistical analysis done over the results presented in Fig. 3. The upper diagonal shows the p value of the results that were significantly different. The lower diagonal shows the values of Cohen's d effect size.

	AIRP	AIRP IDF	MIRP	MIRP IDF	Random
AIRP		0.017	-	-	4.4×10^{-4}
AIRP IDF	0.230		-	-	1.2×10^{-3}
MIRP	-	-		-	5.4×10^{-4}
MIRP IDF	-	-	-		1.5×10^{-4}
Random	0.316	0.344	0.309	0.339	

6. Results

In this section, we present the results regarding the experiments defined in Section 5. Every result presented in a graph includes its respective 95% confidence interval.

6.1. Experiments with No Relevance Feedback

In Fig. 3 we present the results of AIRP and MIRP with and without the Inverse-Document Frequency (IDF). We also compare the results with respect to the random baseline.

As it can be seen in Fig. 3, all the methods presented in this work surpass the value given by the random baseline. Nonetheless, AIRP and MIRP get similar MAP values.

The corresponding rANOVA between the results presented in Fig. 3 indicates that there is a significant difference between the results (p value $= 2.153 \times 10^{-5}$). According to the *post hoc* test all the methods are significantly different with respect to the random baseline (p value < 0.001). Moreover, AIRP with IDF is significantly different to AIRP (p value $= 0.017$). For the remaining pairs of methods, there is no statistical difference. The average effect size between our methods with respect to the random baseline is $d = 0.327$, which is medium-small. The effect size between AIRP and AIRP with IDF is $d = 0.230$. In Table 2, we present a summary of the results from the statistical test.

In Fig. 4, we compare again our methods with respect to a random baseline but also with the one based on the similarity between job offers and résumés. This experiment was done uniquely over the corpus' subset composed of 60 job posting for which we had found their respective job offers (see Section 4).

As shown in Fig. 4, our methods rank the résumés better than methods using the similarity between job offers and résumés. Moreover, our methods work better on these 60 job postings than with the complete set of 171. The reasons for these results will be discussed in Section 7.

The rANOVA performed on the results shown in Fig. 4 indicates that there was a significant difference between the meth-

[21] The only exception could be LO-MATCH, which provided a service through a website during a time. However, the software, *per se*, was never available to download for testing purposes.

[22] Apart from the AP, we can find in the literature two other metrics specialized in the evaluation of rankings: Kendall's tau and (Normalized) Discounted Cumulative Gain (Järvelin & Kekäläinen, 2000). These metrics are used in rankings with multiple grades of relevance, e.g., *very relevant, relevant, irrelevant* and *very irrelevant*. However, our data set is only annotated with two grades of relevance, thus, AP is the most appropriate metric.

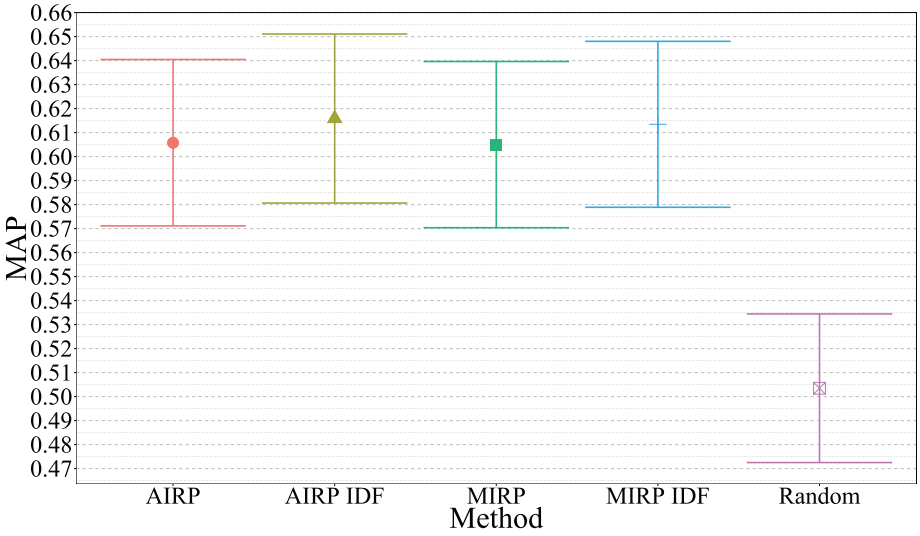

Fig. 3. Results, in terms of the MAP, for the random baseline, AIRP and MIRP without applying any kind of Relevance Feedback.

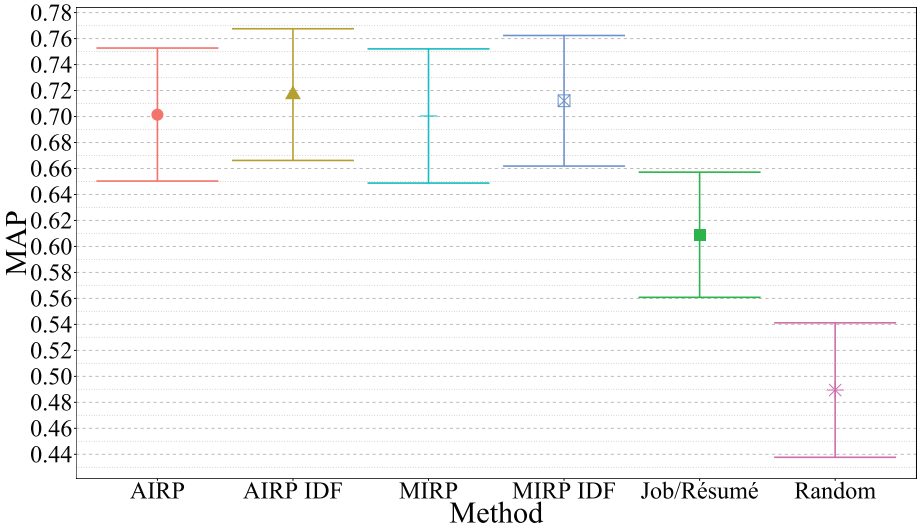

Fig. 4. Comparison of our methods and two baselines (random and similarity between job offer and résumés) for 60 job postings. The values are presented in terms of the MAP, and we did not use any kind of Relevance Feedback.

Table 3
Summary of the statistical analysis done over the results presented in Fig. 4, which correspond to the subset of 60 job postings. The upper diagonal shows the p value of the results that were significantly different. The lower diagonal shows the values of Cohen's d effect size.

	AIRP	AIRP IDF	MIRP	MIRP IDF	Job Offer/Résumé	Random
AIRP		0.045	-	-	7.8×10^{-7}	3.6×10^{-5}
AIRP IDF	0.357		-	-	1.5×10^{-9}	7.8×10^{-6}
MIRP	-	-		-	1.2×10^{-6}	4.7×10^{-5}
MIRP IDF	-	-	-		4.1×10^{-9}	1.1×10^{-5}
Job Offer/Résumé	0.800	1.012	0.784	0.977		0.025
Random	0.656	0.716	0.701	0.642	0.392	

ods (p value $= 3.270 \times 10^{-7}$). In fact, and in accordance with *post hoc* test, the method based on the similarity of job offer/résumé is significantly different than the random baseline and all our methods (p value < 0.05). The effect size between the methods AIRP IDF, MIRP and MIRP IDF, and the job offer/résumé baseline is always $d > 0.780$, which correspond to large effect sizes. In Table 3, we present a summary of the results from the statistical test.

6.2. Experiments with Relevance Feedback

In this part, we present the experiments run with Relevance Feedback (Section 3.3) and applied using two different methods, Relevance Factor and Vocabulary Scoring. It should be noted that as there is no significant difference between AIRP and MIRP, we excluded from the following experiments MIRP. We decided to use

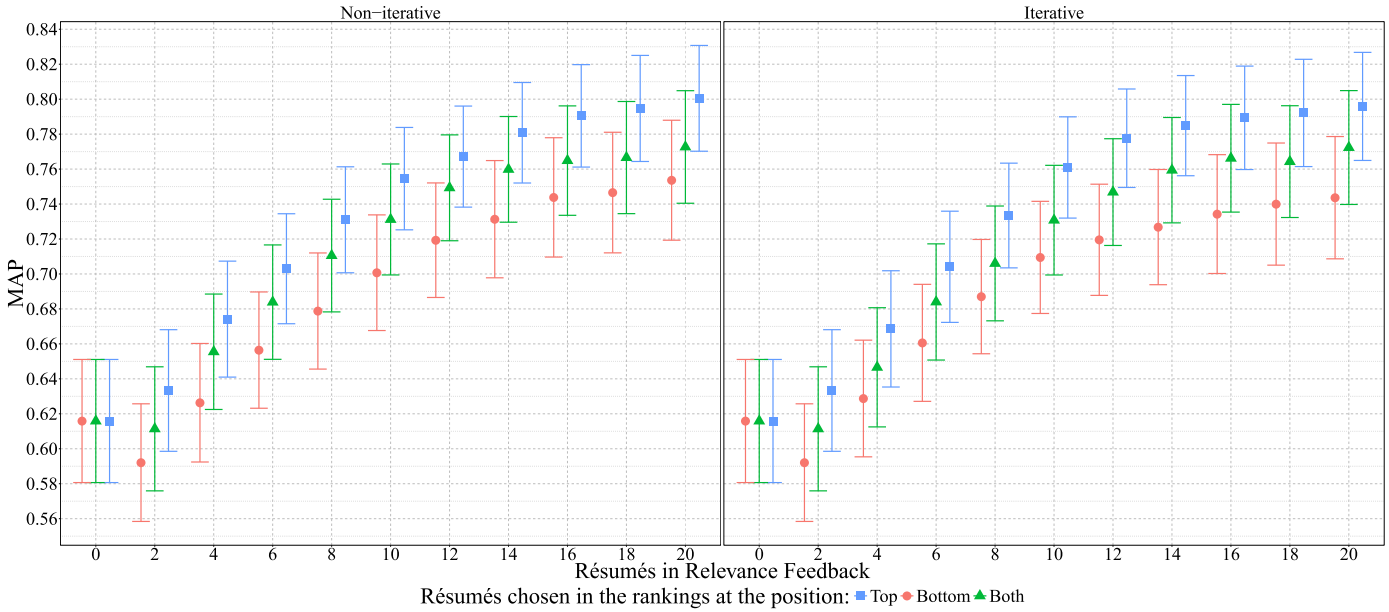

Fig. 5. Results of AIRP IDF after the introduction of the Relevance Feedback using the Relevance Factor. We present, as well, the performance depending on the different positions from where we could obtain the résumés for the Relevance Feedback.

Table 4
Summary of the statistical analyses, done over the results, at 10 and 20 résumés, regarding the non-iterative Relevance Factor. The upper diagonal shows the p value of the results that were significantly different. The lower diagonal shows the values of Cohen's d effect size.

	10 résumés			20 résumés		
	Top	Bottom	Both	Top	Bottom	Both
Top		2.1×10^{-10}	1.7×10^{-4}		5.8×10^{-9}	5.0×10^{-7}
Bottom	0.532		2.3×10^{-5}	0.574		1.4×10^{-3}
Both	0.293	0.345		0.484	0.290	

uniquely the AIRP IDF because it showed a statistical difference with AIRP, moreover, in real cases the IDF could be of help in reducing the n-grams that are frequent but useless for HRM.

6.2.1. Relevance Factor

We present the results regarding the Relevance Factor and how the Relevance Feedback positions (Top, Bottom and Both) affected its performance. Furthermore, we verified whether the iterative application of the Relevance Feedback could improve the speed of résumé ranking. In each iterative step 2 résumés were analyzed until reveal up to 20 résumés. The results of these experiments are presented in Fig. 5.

In Fig. 5, we see that the Relevance Feedback depends on where the résumés are obtained: Top, Bottom or Both positions. The Top position needs a smaller number of résumés to generate higher values of MAP than the Bottom position does.

The rANOVA done with 10 and 20 résumés indicated a significant difference between the positions in the non-iterative process, p value $= 2.45 \times 10^{-12}$ and p value $= 6.35 \times 10^{-11}$ respectively. More specifically, the pairwise *post hoc* test revealed that there was always a significant difference with 10 and 20 résumés for all the Relevance Feedback positions (p value < 0.005). In Table 4, we present a summary of the statistical analyses and the effect sizes obtained. It should be noted that the effect sizes are between medium-small and medium. Similar results for Relevance Feedback positions were obtained with the rANOVA and *post hoc* test for the iterative process.

It can be seen, in Fig. 5, that the iterative application of the Relevance Feedback does not bring any improvement with respect to the non-iterative application. There are some minimal variations, positive or negative, but in most cases the values are the same. In fact, we determined through a rANOVA that there is no significant difference between the iterative and non-iterative application of the Relevance Feedback (p value > 0.05) for 10 and 20 résumés. We can say that both kinds of applications give comparable results. Thus, in the following experiments we use only the non-iterative process.

6.3. Relevance Factor with Vocabulary Scoring

For the Relevance Factor with Vocabulary Scoring, we made use of AIRP IDF with a non-iterative Relevance Feedback application. Vocabulary Scoring was done following simulations S_1, S_2 and S_3, as explained in Section 4.2. In Fig. 6, we present the results from these experiments.

We see from Fig. 6 that the results in terms of the MAP depend on the simulation utilized for Vocabulary Scoring. On one hand, it is evident that simulation S_3, where we used the maximum quantity of information available to calculate the Term Scores, completely boosts the Relevance Factor and allows us reaching a MAP of 0.937 ± 0.014. On the other hand, simulations S_1 and S_2 do not improve the Relevance Factor. It can be seen in Fig. 6 that S_1, despite being conceived to boost the n-grams that represented the classes, relevant and irrelevant, reduces the performance of the Relevance Factor in comparison to its application without Vocabulary Scoring. For instance, using 20 résumés in the Relevance Feedback process without any Vocabulary Scoring results in the MAP being equal to 0.800 ± 0.030, while using simulation S_1 results in

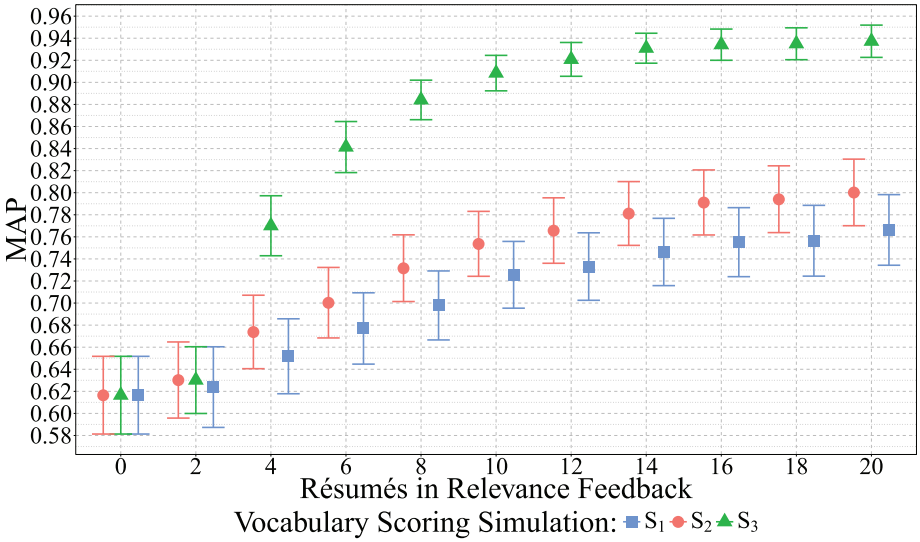

Fig. 6. Results of AIRP IDF using a Relevance Feedback that was applied with the Relevance Factor and Vocabulary Scoring. The Vocabulary Scoring was obtained through three different simulation S_1, S_2 and S_3.

Table 5
Summary of the statistical analyses, done over the results, at 10 and 20 résumés, regarding the application of Relevance Factor with three simulations of Vocabulary Scoring (S_1, S_2, S_3). The upper diagonal shows the p value of the results that were significantly different. The lower diagonal shows the values of Cohen's d effect size.

	10 résumés			20 résumés		
	S_1	S_2	S_3	S_1	S_2	S_3
S_1		1.2×10^{-8}	$< 2 \times 10^{-16}$		6.2×10^{-14}	$< 2 \times 10^{-16}$
S_2	0.458		$< 2 \times 10^{-16}$	0.749		$< 2 \times 10^{-16}$
S_3	0.978	0.858		1.163	1.013	

a MAP value of 0.766 ± 0.031. In contrast, in S_2, where we do not consider the representative n-grams of each class, the MAP stayed stable as if the Vocabulary Scoring would have not been used. This outcome, will be discussed in Section 7.

The rANOVA performed on the results showed there was a significant difference between the simulations using 10 and 20 résumés, in both cases p value $= 2.2 \times 10^{-16}$. According to the pairwise *post hoc* test, at 10 and 20 résumés, all the simulations were significantly different. In Table 5, we present the results regarding p value and effect size.

Regarding the effect size, at 10 résumés, between simulation S_1 and S_2 Cohen's $d = 0.458$, which is large-small; between S_3 and, S_1 and S_2, Cohen's d was greater than 0.850, which it is a large effect size. Using 20 résumés, the effect size between S_1 and S_2 was large-medium effect size $d = 0.749$, for the rest of pairs, Cohen's d was greater than 1, which correspond to a large effect size.

7. Discussion

In the following subsections, we discuss the results obtained in Section 6. The discussion is divided based on the experiments.

7.1. AIRP, MIRP, IDF and baselines

The significant difference between our methods and the random baseline method means that our methods can be, by themselves, of help to HRMs. In other words, the Inter-Résumé Proximity, used through AIRP and MIRP, can rank correctly, to a certain degree, the résumés and proposes a better start point, than a random one, to HRMs during the selection process. As we observed in Section 6.1, there was no significant difference between AIRP and MIRP. This finding means that the distribution of Inter-Résumé Proximities is often symmetrical and does not contain outliers.

We observed that between all our methods and the random baseline there was a statistical difference, however between our other methods, in general, there was not a significant difference. Moreover, the rANOVA performed on the results presented over the subset of 60 job postings (Fig. 4) suggests that our methods are better than the method based on the similarity between job offers and résumés. We could see this, as evidence that résumés contain more information about the job requirements than the job offer does, at least without using semantic resources. This could also mean that the vocabulary used in the job offer and the résumés differs to a certain degree.

It is interesting how in terms of MAP, our methods worked better over the 60 job postings to which we had access to the job offer than for the set of 171 job postings. One reason for this outcome might be that these 60 job postings had one particular characteristic: on average, the number of relevant résumés was 2.2 times the number of irrelevant résumés. This contrasts with the average number of relevant résumés for the 171 job postings, which was 1.4 times the number of irrelevant résumés. Another explanation, is that this difference can be a signal that the "true" MAP, the one that would be obtained if we analyze the statistical population instead of a statistical sample, is located between 0.60 and 0.73. Although these could be the main reasons, we do not leave aside the fact that there could be others, intrinsic or not, to these job postings. To find these other reasons, we need to perform a deeper analysis of these job postings and validate whether the number of relevant résumés had an impact on the performance of AIRP and MIRP.

7.2. Relevance Feedback positions and the Relevance Factor

As we observed in Section 6.2.1, the Relevance Factor is affected by the place from where the résumés used for the Relevance Feedback were obtained. In fact, the most helpful position was the Top one while the Bottom position was the one that gave the lowest performance. The latter result indicates that at the end of the rankings we did not find relevant résumés. In other words, we do not find résumés that could help us determine what is sought by the HRM. As a consequence, it is difficult to improve the results using only irrelevant résumés. Moreover, in order to see an improvement

with the Bottom position, it is necessary to increase the number of analyzed résumés. This means reaching the middle of the rankings, from the bottom, to increase the probability of finding relevant résumés.

Despite the Both position results were less performing than those obtained with the Top position, it could be of interest to follow it in real life. The main reason is that it may verify that we did not leave someone relevant at the end of the résumé ranking. The second reason is that its behavior is not far from the behavior obtained with the Top position; although according to the statistical test, there is a significant difference and the effect size is between small and medium-small.

It is of interest to determine whether an asymmetric Both position is better than a symmetric one. Currently, the same number of résumés is analyzed from the top and the bottom of the résumé rankings. However, it may be better to analyze more résumés from the top of the rankings than from the bottom to improve the speed of our methods.

It can be asked why the MAP decreases when using two résumés for Relevance Feedback for the Bottom and Both positions. The reason is that we increase the probability of finding only irrelevant résumés by looking for résumés at these positions. When we use only irrelevant résumés for the Relevance Factor (Eq. (6)), we can penalize relevant résumés based on their small similarities with the irrelevant ones. As mentioned previously, by increasing the number of analyzed résumés, we can increase the number of relevant résumés analyzed and reduce the effect of the irrelevant ones located at the end of the rankings.

We did not found any significant difference with respect to the iterative and non-iterative application of the Relevance Feedback. Moreover, we do not have a precise idea of why the iterative application did not improve the speed of résumé ranking. The best idea that we have is that the improvement is so small that the MAP cannot detect it. Put differently, the résumés just change ranking positions with other résumés of the same type (relevant or irrelevant) and this cannot be detected by the MAP. It is possible that the number of résumés used in each iteration, two, is not enough to provide visible improvement. We may need to determine with other experiments how many résumés are necessary in an iterative application of the Relevance Feedback to see real improvement.

To improve the performance of the iterative application of the Relevance Feedback, we may need as well to take into account the history of how the résumés move within the rankings. If we find that résumé rankings do not change greatly, it could mean that we arrived at a point where we cannot further improve the rankings with this method. Thus, we should change the method, for example, by using Vocabulary Scoring or looking for relevant résumés at the bottom, or even at a random position.

7.3. Vocabulary Scoring

The results obtained using Vocabulary Scoring and the Relevance Factor were surprising. We never expected to surpass a MAP of 0.9, as we did with S_3 (MAP of 0.9372 ± 0.014). Furthermore, we were surprised by the results because Vocabulary Scoring only affects the model used in determining the Relevance Factor. Thus, the AIRP of one résumé r is modified only by the Relevance Factor (Eq. (6)) which determines how proximal résumé r is to the relevant and irrelevant ones using basically 100 n-grams chosen by the HRM (50 terms per class).

The poor performance of S_1 and S_2, seen in Section 6.3, may be related to the quantity of data utilized to establish the Term Scores. Using only the information provided by documents from the Relevance Feedback is not enough to simulate correctly the knowledge that an HRM would have about the job posting and, in consequence, to determine the Term Scores. It should be remembered that the simulations are based on the squared probabilities (Eq. (9)) and without enough information these values lack the reliability to correctly represent the classes. Although, we tried to increase the reliability by using only n-grams observed in at least two résumés, as explained in Section 4.2.1, this minimum might not be enough for these two simulations. The problem is solved when we make use of S_3, where we calculate the squared probabilities based on all the information available.

To better understand how the simulations worked and affected the results, we present in the following lines a discussion of the simulations generated regarding a *Project Manager* job posting; this job posting is one of the 60 job postings linked manually to the job offer. In Fig. 7, we present an abstract of the job offer related to the job posting. In Table 6, we present an extract of Vocabulary Scoring using the three simulations, S_1, S_2 and S_3, for 20 résumés of Relevance Feedback.[23] It should be remembered, that for obtaining the n-grams and the values presented in Table 6, we did not make use of the job offer at any moment, they are result from simulation S_1, S_2 and S_3 as explained in Section 4.2.1.

We see from Table 6 that simulation S_3 provides the best weights to the terms related to the job offer, even when the last one was not included in the analysis process. Nevertheless, S_1 and S_2 have trouble correctly weighting the terms of the job offer or at least placing them within the first five positions; the reason is the lack of information.

Additionally, although impossible to show due to their length, it should be mentioned that for simulation S_3, the n-grams of both classes always had a squared probability, $p_c^2(t)$, of 1. For simulations S_1 and S_2 the squared probabilities were always 1 regarding the relevant class, while they varied from 1 to 0.444 for the irrelevant class.

In general, thanks to outputs like those presented in Table 6, it is possible to better understand which characteristics were the ones looked for or impacted the decision of HRM. With this kind of lists, psychologist can do *a posteriori* studies regarding the selection of candidates. Or, other HRMs can use this kind of output to explain to candidates why they were not selected for an interview.

One interesting thing to note, as seen in Fig. 6, is that S_2 is better than S_1 despite the former did not contain the terms that were boosted in the latter. The reason for this discrepancy is related to the quality of the n-grams chosen for the simulations and how we determine the Term Scores. As seen in Table 6, the terms used for simulations S_1 and S_2, especially those for the relevant résumés, are quite different from the terms found in S_3 and in the job offer. They can be considered as "bad" in terms of representativeness. Thus, in S_1 we gave these "bad" n-grams the power to reflect the classes, even though they do not truly represent them; the consequences are bad rankings. In S_2 we deleted these "bad" terms, while the rest of terms represented the classes, although with poor Term Scores; the resulting rankings are affected negatively but not as much as in S_1.

In the previous results, we can see that the terms chosen by HRM may have a crucial role in the performance of Vocabulary Scoring, and as a consequence on the performance of the Relevance Factor. In other words, to choose terms that do not correctly represent what an HRM wants and does not want can negatively impact the ranking of résumés.

Related to this last point, we want to know how the Vocabulary Scoring is affected by the way the terms are sorted because it may not be an obvious task for an HRM to perform. In fact, an HRM

[23] Simulations S_1 and S_2 sort in the same way the n-grams; their difference is that S_2 gives a Term Score of 0 to the first 50 n-grams. Simulation S_3 makes use of all the information available in the job to sort the terms presented in the 20 résumés analyzed.

We look for a project manager for rail development. He/she must assure the product's quality, price and timing. The person should be an engineer or a Ph.D. with a specialization on mechanics. He/she must have at least 5 year of experience in the industry (e.g., automobile, rail, aeronautics or mechanical transmissions). Knowledge of the rail sector will be appreciated.

Fig. 7. Summary of a Project Manager job offer. The job offer comes from one of the 60 job postings to which we found their respective job offers. The original job offer was in French; we translated it to English and summarized it.

Table 6
Squared probabilities, sum of weights, number of documents, factors and rank for a set of terms according to each Vocabulary Scoring simulation. All the *n*-grams, originally in French but translated to English, belong to the résumés linked to the job offer presented in Fig. 7. The job has in total 36 relevant résumés and 29 irrelevant ones.

Simulations	Class	*n*-gram (t)	$p_c(t)$	$\Sigma W(t)$	$D_c(t)$	$f_c(t)$	Rank
S_1 and S_2	Irrelevant	Project engineer	1	0.024	3	0.072	1
		Micro-techniques	1	0.022	2	0.045	2
		Investment	1	0.013	3	0.040	3
		SolidWorks Catia V5	1	0.019	2	0.039	4
		Supplier France	1	0.019	2	0.039	5
	Relevant	Business	1	0.040	7	0.285	1
		Rail	1	0.030	7	0.216	2
		Planning	1	0.024	8	0.196	3
		Range	1	0.023	8	0.189	4
		Respect	1	0.024	7	0.174	5
S_3	Irrelevant	Responsible supplier	1	0.038	4	0.154	1
		Unit	1	0.026	4	0.106	2
		Renault project	1	0.032	3	0.098	3
		To orient	1	0.024	4	0.096	4
		Validation piece	1	0.041	2	0.083	5
	Relevant	Rail	1	0.023	22	5.098	1
		Alstom transport	1	0.074	8	0.598	2
		Train	1	0.076	7	0.532	3
		TGV	1	0.062	6	0.372	4
		CAD software	1	0.048	5	0.241	5

can ask how to determine whether one term better represents the relevant or irrelevant résumés than another one. Moreover, they can question whether to "incorrectly" sort one term would affect the resulting ranking at the same level as choosing a bad term. To answer these questions, instead of computing the Term Score with Eq. (7), we decided to assign a Term Score of 1 to the 50 more representative *n*-grams of each class. This is equivalent to saying that the order in which the *n*-grams are sorted has no importance.

The results of setting the Term Scores equal to 1 using simulation S_3 showed that at 10 résumés, we get a MAP of 0.913 ± 0.015; at 20 résumés, the MAP is 0.947 ± 0.012. The rANOVA between our method using Term Scores set to 1 and those computed with the 5^{th} root indicated there is no significant difference at 10 and 20 résumés (p value $= 1.7 \times 10^{-3}$ and p value $= 1.37 \times 10^{-9}$ respectively). These outcomes do not mean that both methods are equivalent and as a consequence interchangeable, but that they perform very similarly.[24] As well, the results obtained from using a Term Score of 1 may provide a hint that the success of Vocabulary Scoring is related more to the quality of the chosen *n*-grams and the weight difference we create with respect to the other terms, i.e., those to which we set a Term Score of 0.01. In other words, to put the most representative *n*-gram at the 50^{th} position of the Vocabulary Scoring does not affect the results as much as leaving it aside.

One interesting thing we observed in five different job postings using S_3 is that the top ranked n-grams from the relevant résumés appear in more documents than the top ranked *n*-grams from the irrelevant résumés. We see this behavior in column $D_c(t)$ of Table 6. If this is true for all the job postings, we could confirm the ideas on which we based AIRP and MIRP: the résumés from relevant applicants have in common multiple terms while the irrelevant résumés usually present a great variety of terms that are not frequently shared. However, we must perform a deeper analysis to validate this hypothesis.

Despite the interest to determine what would be the results using human judgments instead of simulations, it should be noted that this cannot be done without redoing the selection process. The main reason is the relation between the selection of applicants and the person specification, a document that can evolve over time. In other words, the HRM who would redo the selection process may not have access to the previous person specification. This may result in a different evaluation of résumés, especially those from the first candidates who applied. However, we can imagine that in reality, humans would do a good job, even better than simulations, because they know *a priori* the person specification.

Although we did not test Vocabulary Scoring with a set of less than 50 *n*-grams, it may be possible to reduce this figure. In first place, we should test whether a smaller Vocabulary Scoring with Term Scores set to 1, or determined by Eq. (7), have the same performance. If this is not the case, we may change Eq. (7). For example, a gradient closer to zero might help to give better results to the top 10 terms. Another option would be to further reduce the Term Score for the *n*-grams that do not appear in the Relevance Feedback. In previous experiments, not presented here, we observed that as we decreased the Term Scores of the unseen *n*-grams the results were boosted even more.

Moreover, it could be of help to find the *n*-grams or terms, and even their synonyms, that appear in the job offer and person specification in order to improve or automate the generation

[24] The lack of significant difference between two means does not express that they are equal. It indicates that we need more data to determine a significant difference. However, the effect size of this difference may be very small and, in consequence, they would behave very similar in real conditions.

of Vocabulary Scorings. In other words, these *n*-grams or terms could be those that should be positioned at the top of the Vocabulary Scoring. To this end, we could make use of Human Resources lexica, ontologies and terminological extractors. However, the use of these resources may introduce some difficulties as terms may not correspond exactly to the *n*-grams used in the vector model.

8. Conclusions and future work

The massive access of the Internet has changed multiple aspects of our lives, and the way we find and apply for a job offer is not an exception. Although the use of computers and the Internet has made easier to find job offers and potential candidates to send their résumés or curricula vitae, it has negatively affected the performance of human resource managers during the selection process. Human resource managers have trouble to find rapidly the candidates, among all who applied, that meet the job requirements and should be called for an interview.

We presented two innovative methods for ranking résumés by relevance, making it easier for human resource managers to identify candidates with the desired characteristics. The methods here presented are innovative because they make use only of the résumés sent in response to a job offer. These methods contrast with state-of-the-art methods that usually compare résumés and job offers with proximity measures. Our methods are language independent and do not need semantic resources to work. Moreover, the methods presented here are statistically better than a random baseline or a baseline grounded on the similarity between résumés and a job offer.

Moreover, we presented two different ways to apply Relevance Feedback in a résumé ranker. One method for applying Relevance Feedback works at a general level (Relevance Factor), while the other method works at a finer lexical one (Vocabulary Scoring). Although the Relevance Factor helps to improve résumé rankings, we find that it is its use along with Vocabulary Scoring that helps us to reach a Mean Average Precision of 0.937. Put differently, by using the Relevance Factor with Vocabulary Scoring we can correctly rank almost every résumé. As a consequence, we can reduce the time needed by human resource managers to find the résumés of relevant applicants. It is important to note that the very good results obtained with Vocabulary Scoring reinforces the concept that relevant résumés share more characteristics with themselves than with irrelevant ones, as seen in our previous works.

We believe that, within the résumés we can intrinsically find a "facial composite" of the ideal candidate, and possibly the "facial composite" that represents the unqualified candidates. It may be these "facial composites" that enable us to rank résumés without the use of a job offer or semantic resources.

We consider that methodologies based only on résumés and their vocabularies are the future of résumé rankers. The main reason to think this is that they are capable of offering excellent performance without being limited to one domain or language. Despite these methods were created to be used in a particular database, where it was impossible to have access to every job offer, we believe that it can be used in any database of résumés, only if these are separated by job postings. Furthermore, the methods here presented do not make use of any kind of semantic resources, which can make them easier to implement in under-resources languages.

There are still things that must be studied with this kind of methods. In the first place are the temporal aspects. We assumed in this article that all the résumés were present at the same time, but in real life this may not be true. On occasions, the process of recruitment and selection are done in parallel, i.e., once a résumé arrives to a human resource manager, it is analyzed. We have to consider as well the evolution of the person specification over time. In some cases, human resource managers are obliged to become more or less strict in order to filter the applicants. These changes, in consequence, will affect the human resource managers' perception regarding the relevance of applicants. Due to this effect, the way to apply our methods may need to change, and we should evaluate until which extent they remain valid. However, despite all, the proposed methods could be used to evaluate *a posteriori* the reasons why a group of candidates was chosen to do an interview. Moreover, other human resource managers or psychologists may find useful the tool to determine whether human resource managers were affected by personality inferences, misspellings or any kind of discrimination.

Another aspect to take into account is the way to match terms or concepts and *n*-grams. These representations are not the same, and this can infuse difficulty to some degree in the application of our methods. Put differently, a concept may be difficult to represent with an *n*-gram. Finally, it should be analyzed the economics and whether human resource managers will adopt these methods to make their tasks easier.

Regarding the scalability of the methods here presented, we do not observe any particular problem. As we indicated in Section 5, the methods were called using the program *GNU Parallel*, meaning that each job posting was analyzed using different CPU threads. This indicates that multiple job postings can be processed at the same time without any collision. Furthermore, it is possible to parallelize the similarity between résumés, i.e., to use several threads to calculate multiple Dice's Coefficient scores at the same time. The only aspect to take into consideration is that the vectors representing the résumés should be accessible to every thread. At the end, all the methods described in this work can be easily scaled and distributed in a cluster.

In the future, we would like to use *word embedding* in order to calculate the proximity between résumés differently. It could also be useful for Vocabulary Scorings. In addition, we will work on the improvements described in the discussion. Since the methods developed here are language independent, it will be easy to test them on other languages than French. Although this last task can be difficult to achieve due to the lack of a corpus of real selection processes. During the experimentation, we observed that our methods can keep a good performance when they are tested on an encrypted version of the data set here used.[25] Therefore, we can rely on this clue that for other languages, the methods should work as well.

In conclusion, we hope that our methods and results will attract new and deeper research in this domain.

Credit authorship contribution statement

Luis Adrián Cabrera-Diego: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Marc El-Béze:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing - review & editing, Supervision, Project administration, Funding acquisition. **Juan-Manuel Torres-Moreno:** Conceptualization, Methodology, Writing - review & editing, Supervision, Project administration, Funding acquisition. **Barthélémy Durette:** Conceptualization, Methodology, Formal analysis, Writing - review & editing, Supervision, Project administration.

[25] We did not achieve the same results in the encrypted data set, as the résumés were encrypted without doing a deep pre-processing, like lemmatization or stop words deletion. Thus, the résumés contained a greater variety of terms and noisy words.

Acknowledgments

This work was partially funded by the *Agence National de la Recherche et de la Technologie* (ANRT), France, through the CIFRE convention 2012/0293b and by the *Consejo Nacional de Ciencia y Tecnología* (CONACyT), Mexico, with the grant 327165.

References

Armstrong, M., & Taylor, S. (2014). *Armstrong's handbook of human resource management practice* (13th). Kogan Page Publishers.

Arthur, D. (2001). *The employee recruitment and retention handbook*. AMACOM.

Barber, L. (2006). *E-Recruitment developments*. Institute for Employment Studies.

Buckley, C., & Voorhees, E. M. (2000). Evaluating evaluation measure stability. In *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 33–40). Athens, Greece: ACM. doi:10.1145/345508.345543.

Cabrera-Diego, L. A. (2015). *Automatic methods for assisted recruitment*. Université d'Avignon et des Pays de Vaucluse Ph.D. thesis.

Cabrera-Diego, L. A., Durette, B., Lafon, M., Torres-Moreno, J.-M., & El-Bèze, M. (2015). How can we measure the similarity between résumés of selected candidates for a job?. In Stahlbock, Robert, & Weiss, Gary M. (Eds.), *Proceedings of the 11th international conference on data mining (DMIN'15)* (pp. 99–106). Las Vegas, USA

Chapman, D. S., & Webster, J. (2003). The use of technologies in the recruiting, screening, and selection processes for job candidates. *International Journal of Selection and Assessment, 11*(2–3), 113–120. doi:10.1111/1468-2389.00234.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd). Hillsdale, USA: Lawrence Earlbaum Associates.

Cole, M. S., Feild, H. S., Giles, W. F., & Harris, S. G. (2009). Recruiters' inferences of applicant personality based on résumé screening: Do paper people have a personality? *Journal of Business and Psychology, 24*(1), 5–18. doi:10.1007/s10869-008-9086-9.

Cossu, J.-V. (2015). *Analyse de l'image de marque sur le Web 2.0*. Avignon, France: Université d'Avignon et des Pays de Vaucluse Ph.D. thesis.

Cossu, J.-V., Janod, K., Ferreira, E., Gaillard, J., & El-Bèze, M. (2014). LIA@RepLab 2014: 10 methods for 3 tasks. In L. Cappellato, N. Ferro, M. Halvey, & W. Kraaij (Eds.), *Working notes for 4th International Conference of the CLEF initiative* (pp. 1458–1467). Sheffield, UK

Elkington, T. (2005). Bright future for online recruitment. *Personnel Today*, 9.

Faliagka, E., Iliadis, L., Karydis, I., Rigou, M., Sioutas, S., Tsakalidis, A., & Tzimas, G. (2013). On-line consistent ranking on e-recruitment: Seeking the truth behind a well-formed CV. *Artificial Intelligence Review*, 1–14. doi:10.1007/s10462-013-9414-y.

Faliagka, E., Kozanidis, L., Stamou, S., Tsakalidis, A., & Tzimas, G. (2011). A personality mining system for automated applicant ranking in online recruitment systems. In S. Auer, O. Díaz, & G. A. Papadopoulos (Eds.), *Proceedings of the 11th international conference web engineering (ICWE 2011)*. In *Lecture Notes in Computer Science: 6757* (pp. 379–382). Paphos, Cyprus: Springer Berlin Heidelberg. doi:10.1007/978-3-642-22233-7_30.

Fang, X., & Zhan, J. (2015). Sentiment analysis using product review data. *Journal of Big Data, 2*(1), 5. doi:10.1186/s40537-015-0015-2.

García-Sánchez, F., Martínez-Béjar, R., Contreras, L., Fernández-Breis, J. T., & Castellanos-Nieves, D. (2006). An ontology-based intelligent system for recruitment. *Expert Systems with Applications, 31*(2), 248–263. doi:10.1016/j.eswa.2005.09.023.

Guo, S., Alamudun, F., & Hammond, T. (2016). RésuMatcher: A personalized résumé-job matching system. *Expert Systems with Applications, 60*(Supplement C), 169–182. doi:10.1016/j.eswa.2016.04.013.

Harzallah, M., Leclère, M., & Trichet, F. (2002). CommOnCV: Modelling the competencies underlying a curriculum vitae. In *Proceedings of the 14th international conference on software engineering and knowledge engineering (SEKE'02)* (pp. 65–71). Ischia Island, Italy: ACM. doi:10.1145/568760.568773.

Hutterer, M. (2011). *Enhancing a job recommender with implicit user feedback*. Vienna, Austria: Fakultät für Informatik der Technischen Universität Wien Master's thesis.

Järvelin, K., & Kekäläinen, J. (2000). IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 41–48). Athens, Greece: ACM. doi:10.1145/345508.345545.

Kessler, R., Béchet, N., Roche, M., El-Bèze, M., & Torres-Moreno, J. M. (2008a). Automatic profiling system for ranking candidates answers in human resources. In R. Meersman, Z. Tari, & P. Herrero (Eds.), *On the move to meaningful internet systems: OTM 2008 Workshops*. In *Lecture Notes in Computer Science: 5333* (pp. 625–634). Monterrey, Mexico: Springer Berlin Heidelberg. doi:10.1007/978-3-540-88875-8_86.

Kessler, R., Béchet, N., Roche, M., Torres-Moreno, J.-M., & El-Bèze, M. (2012). A hybrid approach to managing job offers and candidates. *Information Processing & Management, 48*(6), 1124–1135. doi:10.1016/j.ipm.2012.03.002.

Kessler, R., Béchet, N., Torres-Moreno, J.-M., Roche, M., & El-Bèze, M. (2009). Job offer management: How improve the ranking of candidates. In *Foundations of intelligent systems: Proceedings of 18th international symposium on methodologies for intelligent systems (ISMIS 2009)*. In *Lecture Notes in Computer Science: 5722* (pp. 431–441). Prague, Czech Republic: Springer Berlin Heidelberg. doi:10.1007/978-3-642-04125-9_46.

Kessler, R., Torres-Moreno, J. M., & El-Bèze, M. (2008b). E-Gen: Profilage automatique de candidatures. In *Actes de la 15ème conférence sur le Traitement Automatique des Langues Naturelles (TALN 2008)* (pp. 370–379). Avignon, France

Kmail, A. B., Maree, M., & Belkhatir, M. (2015). MatchingSem: Online recruitment system based on multiple semantic resources. In *12th international conference on fuzzy systems and knowledge discovery (FSKD 2015)* (pp. 2654–2659). doi:10.1109/FSKD.2015.7382376.

Looser, D., Ma, H., & Schewe, K.-D. (2013). Using formal concept analysis for ontology maintenance in human resource recruitment. In F. Ferrarotti, & G. Grossmann (Eds.), *Proceedings of the ninth Asia-Pacific conference on conceptual modelling: 143* (pp. 61–68). Adelaide, Australia: Australian Computer Society, Inc.

Martin-Lacroux, C. (2017). "Without the spelling errors I would have shortlisted her...":The impact of spelling errors on recruiters' choice during the personnel selection process. *International Journal of Selection and Assessment, 25*(3), 276–283. doi:10.1111/ijsa.12179.

Martinez-Gil, J., Paoletti, A. L., Rácz, G., Sali, A., & Schewe, K.-D. (2018). Accurate and efficient profile matching in knowledge bases. *Data & Knowledge Engineering, 117*, 195–215. doi:10.1016/j.datak.2018.07.010.

Martinez-Gil, J., Paoletti, A. L., & Schewe, K.-D. (2016). A smart approach for matching, learning and querying information from the human resources domain. In M. Ivanović, B. Thalheim, B. Catania, K.-D. Schewe, M. Kirikova, P. Šaloun, A. Dahanayake, T. Cerquitelli, E. Baralis, & P. Michiardi (Eds.), *Proceedings of the new trends in databases and information systems: ADBIS 2016 short papers and workshops, BigDap, DCSA, DC* (pp. 157–167). Prague, Czech Republic: Springer International Publishing. doi:10.1007/978-3-319-44066-8_17.

Mason, R. L., Gunst, R. F., & Hess, J. L. (2003). Statistical design and analysis of experiments: With applications to engineering and science. *Wiley Series in Probability and Statistics* (2nd). Wiley-Interscience. doi:10.1002/0471458503.

Menon, V. M., & Rahulnath, H. A. (2016). A novel approach to evaluate and rank candidates in a recruitment process by estimating emotional intelligence through social media data. In *International conference on next generation intelligent systems (ICNGIS)* (pp. 1–6). Kottayam, India: IEEE. doi:10.1109/ICNGIS.2016.7854061.

Montuschi, P., Gatteschi, V., Lamberti, F., Sanna, A., & Demartini, C. (2014). Job recruitment and job seeking processes: How technology can help. *IT Professional, 16*(5), 41–49. doi:10.1109/MITP.2013.62.

Padró, L., & Stanilovsky, E. (2012). FreeLing 3. 0: Towards wider multilinguality. In N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the eight international conference on language resources and evaluation (LREC'12)* (pp. 2473–2479). Istanbul, Turkey: ELRA.

R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing Vienna, Austria.

Radevski, V., & Trichet, F. (2006). Ontology-based systems dedicated to human resources management: An application in e-Recruitment. In R. Meersman, Z. Tari, & P. Herrero (Eds.), *On the move to meaningful internet systems 2006: OTM 2006 Workshops*. In *Lecture Notes in Computer Science: 4278* (pp. 1068–1077). Montpellier, France: Springer Berlin Heidelberg. doi:10.1007/11915072_9.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), *The SMART retrieval system: Experiments in automatic document processing*. In *Automatic Computation* (pp. 313–323). Englewood Cliffs, N.J., USA: Prentice-Hall.

Salton, G., Wong, A., & Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*(11), 613–620. doi:10.1145/361219.361220.

Sen, A., Das, A., Ghosh, K., & Ghosh, S. (2012). Screener: A system for extracting education related information from resumes using text based information extraction system. In *Proceedings of 2012 international on computer and software modeling (ICCSM 2012)*. In *International proceedings of computer science & information technology: 54* (pp. 31–35). International Association of Computer Science and Information Technology Press (IACSIT Press). doi:10.7763/IPCSIT.2012.V54.06.

Senthil Kumaran, V., & Sankar, A. (2012). Expert locator using concept linking. *International Journal of Computational Systems Engineering, 1*(1), 42–49. doi:10.1504/IJCSYSE.2012.044742.

Senthil Kumaran, V., & Sankar, A. (2013). Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping (EXPERT). *International Journal of Metadata, Semantics and Ontologies, 8*(1), 56–64. doi:10.1504/IJMSO.2013.054184.

Singh, A., Rose, C., Visweswariah, K., Chenthamarakshan, V., & Kambhatla, N. (2010). PROSPECT: A system for screening candidates for recruitment. In *Proceedings of the 19th ACM international conference on information and knowledge management (CIKM 2010)* (pp. 659–668). Toronto, Canada: ACM. doi:10.1145/1871437.1871523.

Spärck-Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28*(1), 11–21. doi:10.1108/eb026526.

Tange, O. (2011). GNU parallel - The command-line power tool. *login: The USENIX Magazine, 36*(1), 42–47.

Thompson, M. A. (2000). *The global resume and CV guide*. Chichester, New York: Wiley.

Tinelli, E., Colucci, S., Donini, F. M., Di Sciascio, E., & Giannini, S. (2017). Embedding semantics in human resources management automation via SQL. *Applied Intelligence, 46*(4), 952–982. doi:10.1007/s10489-016-0868-x.

Torres-Moreno, J.-M., El-Bèze, M., Bellot, P., & Béchet, F. (2012). Opinion detection as a topic classification problem. In É. Gaussier, & F. Yvon (Eds.), *Textual information access: Statistical models* (pp. 337–368). Wiley-ISTE. doi:10.1002/9781118562796.ch9.

Trichet, F., Bourse, M., Leclère, M., & Morin, E. (2004). Human resource management and semantic web technologies. In *Proceedings of information and communication technologies: From theory to applications (ICTTA'04)* (pp. 641–642). Damascus, Syria: IEEE. doi:10.1109/ICTTA.2004.1307928.

Voorhees, E. M., & Harman, D. (2001). Overview of TREC 2001. In *Proceedings of the 10th Text REtrieval Conference (TREC 2001)* (pp. 1–15). Gaithersburg, Maryland, USA: National Institute of Standards and Technology (NIST).

Zaroor, A., Maree, M., & Sabha, M. (2017). A hybrid approach to conceptual classification and ranking of resumes and their corresponding job posts. In I. Czarnowski, R. J. Howlett, & L. C. Jain (Eds.), *Intelligent decision technologies 2017: Proceedings of the 9th KES international conference on intelligent decision technologies (KES-IDT 2017) - part I* (pp. 107–119). Vilamoura, Portugal: Springer International Publishing. doi:10.1007/978-3-319-59421-7_10.