

Matching Jobs and Resumes: a Deep Collaborative Filtering Task

Thomas Schmitt, Philippe Caillou, Michèle Sebag

▶ **To cite this version:**

Thomas Schmitt, Philippe Caillou, Michèle Sebag. Matching Jobs and Resumes: a Deep Collaborative Filtering Task. GCAI 2016 - 2nd Global Conference on Artificial Intelligence, Sep 2016, Berlin, Germany. hal-01378589

HAL Id: hal-01378589
https://inria.hal.science/hal-01378589

Submitted on 13 Oct 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Matching Jobs and Resumes: a Deep Collaborative Filtering Task

Thomas Schmitt, Philippe Caillou, and Michele Sebag

TAO, CNRS − INRIA − LRI, Univ. Paris-Sud
Université Paris-Saclay, France

Abstract

This paper tackles the automatic matching of job seekers and recruiters, based on the logs of a recruitment agency (CVs, job announcements and application clicks). Preliminary experiments reveal that good recommendation performances in collaborative filtering mode (emitting recommendations for a known recruiter using the click history) co-exist with poor performances in cold start mode (emitting recommendations based on the job announcement only). A tentative interpretation for these results is proposed, claiming that job seekers and recruiters − whose mother tongue is French − yet do not speak the same language. As first contribution, this paper shows that the information inferred from their interactions differs from the information contained in the CVs and job announcements.

The second contribution is the hybrid system MAJORE (MAtching JObs and REsumes), where a deep neural net is trained to match the collaborative filtering representation properties. The experimental validation demonstrates MAJORE merits, with good matching performances in cold start mode.

1 Introduction

Among the key problems of the 21^{st} century is unemployment, a multi-faceted societal and industrial phenomenon. This paper considers a single facet thereof, referred to as *informational market labor frictions*: the fact that high unemployment rates can co-exist with high numbers of job unfilled. Machine learning and Knowledge Discovery have been investigating human resource-related applications for over a decade (more in section 2). Focusing more specifically on e-recruitment, the state of the art essentially considers the matching issues related to high paying jobs [2]. A majority of approaches support the recruiter or headhunter task, using natural language processing, statistical or hybrid approaches [13, 6, 15, 9] to identify and rank top management and executive-level candidates with targeted professional skills. Other approaches aim at supporting the seeker's search and retrieving job announcements best fitting her professional skills and goals [19, 24]. Yet other approaches, inspired by social matching and dating [7], handle the seekers and recruiters matching as a reciprocal search [18].

This paper instead considers the matching issues related to temporary and low-wage jobs, tackling the matching of CVs and job announcements and providing support to recruiters. The choice of low-wage jobs is guided by both scientific and applicative motivations. On the applicative side, the needs are huge: the vast majority of job announcements concerns low-wage jobs (according to the French Ministry of Labor, 2016, over 90% staff recruited every month are on temporary, fixed-term contracts). On the scientific side, the challenge is that recruiters and job seekers − whose mother tongue is French − yet do not seem to speak the same language (more in section 3). More precisely, CVs and job announcements tend to use different vocabularies, and same words might be used with different meanings. In other words, the task of accurately matching jobs and CVs might be *more a machine translation problem*, than an information retrieval one. These remarks echo recent advances in information retrieval, also suggesting that

that people tend to use different vocabularies depending on whether they write or search for information [11, 21].

The presented study exploits the logs of a recruitment agency[1] over a 2-month period, including 143,000 job announcements, 65,000 CVs and the usage traces recording which job seeker applied on a job announcement. A first question concerns the quality of the dataset w.r.t. the heterogeneity of the low-wage job labor market, and whether the data enables accurate collaborative filtering, making accurate recommendations for known seekers or recruiters [10, 17]. Job and CV matching however significantly differs from collaborative filtering, as popularized by the Netflix challenge [5]: once a seeker (respectively a recruiter) has found a job (resp. an employee), the search is over, in general. In other words, job and CV matching mostly deals with the cold-start problem of making recommendations for new recruiters or seekers. The second question thus is whether the available textual information, that is, the CV and job announcement documents, enables efficient cold-start recommendation.

The contribution of the proposed paper is twofold. The representativity of the data is established as excellent matching performances can be obtained using collaborative filtering. Nevertheless, poor performances are obtained in cold-start mode. A first contribution regards the interpretation of this counter-performance, blamed on the fact that job seekers and recruiters do not speak the same language. Additional experiments are conducted to back up this claim. The second contribution is the hybrid system MAJORE, using the representation learned by collaborative matching and its properties [20], to train a deep neural net [4]. The merits of MAJORE are empirically shown, with significantly improved cold-start performances compared to baseline methods.

The paper is organized as follows. Section 2 briefly reviews related work concerned with e-recruitment, focusing on the job and CV matching task. The basics of collaborative filtering and singular value decomposition are introduced for the sake of self-containedness. Section 3 presents the data used in the paper. Section 4 gives an overview of the MAJORE system. Experimental settings are reported in section 5 and the empirical validation in section 6. The paper concludes with a discussion and some perspectives for further research.

2 Background knowledge

This section first briefly reviews some machine learning-based approaches related to e-recruitment. Recommendation systems are thereafter introduced for the sake of self-containedness.

2.1 E-recruitment

The state of the art mostly focuses on high paying jobs, exploring several approaches [2]. A first approach is that of Content- and Knowledge Based Recommendation (CKBR) [13, 6, 15]. With roots in Natural Language Processing, CKBR involves the extraction of grammatical, lexical and semantic features from the CVs, application letters and job announcements. These features are possibly augmented using external databases or domain ontologies (e.g. augmenting a job announcement with the skills required for this job). Additionally, similarities are built from statistical models [13] or expert knowledge to compute the degree of matching between jobs and resumes. [15] builds an ontology from the candidate documents and matches it with the job ontology to compute the degree of matching.

Another approach is concerned with the exploitation of new sources of information, e.g. social networks. Firms can examine the applicant profiles (including the restricted parts, e.g.

[1] With kind permission from the Qapa company and its CEO S. Delestre.

asking the applicant's password) [3, 9, 8]. The human perception of job-suitability is correlated with many profile components (e.g., posts, photos, likes). Beyond the education and skill factors, screening based on social networks retains personality-related and other demographic criteria.

Machine learning is a key enabling technology to build scalable systems. In [19], job recommendation is addressed as a supervised machine learning problem, exploiting past job transitions, employee CVs and institution data. In [24], semi-structured supervised ML is used to exploit the structure of the CV and job announcement documents. In [9], the online ranking of candidates is based on features extracted from LinkedIn profiles (including personality features from the linguistic analysis of posts), and yields similar performances as human recruiters. A source of inspiration is that of social recommendation and dating [7], accounting for the two-sided relevance aspects of CVs and job announcements. [18] propose a reciprocal recommendation method, where recruiters and seekers exchange messages, iteratively assessing the popularity of their offer.

2.2 Recommendation approaches

A main trend of e-recruitment is based on adapting and extending recommendation systems [10, 17]. Recommendation systems [23, 1], popularized by the Netflix challenge [5], often exploit the (n, m) matrix $\mathcal{M}$ reporting the feedback of the user community about the products (e.g., items for Amazon or movies for Netflix). Letting n (respectively m) denote the number of users (resp. items), $\mathcal{M}_{i,j}$ describes the ranking of the j-th item by i-th user. Optionally, some description and metrics upon the user and item spaces are available.

Memory-based approaches use the metrics on the user (respectively item) spaces, with convolution from matrix $\mathcal{M}$, to recommend items liked by users similar to the target user (resp. items similar to the items liked by the target user):

$$Recomm(i,j) = \frac{\sum_{i'} Sim(i,i') \cdot \mathcal{M}_{i',j}}{\sum_{i'} \mathcal{M}_{i',j}} \tag{1}$$

Model-based approaches rely on a low-rank decomposition of the CF matrix $\mathcal{M}$, using Singular Value Decomposition (SVD) [22] and variants thereof [14]:

$$\mathcal{M} \approx \mathrm{U} \cdot \mathrm{V}^t \tag{2}$$

where U and V respectively are (n, ℓ) and (m, ℓ) matrices and the rank ℓ of the decomposition is determined by cross-validation. U and V are referred to as the *latent representation* of users and items respectively. The decomposition enables to fill the full matrix (with $\mathcal{M}_{i,j} \approx \langle U_i \cdot V_j \rangle$), thus achieving collaborative filtering (recommending items to known users). The limitation is that the latent representation is unknown for new items or users, preventing its usage to achieve cold start recommendation.

The e-recruitment approach proposed by Malinowski et al [17], extending the pioneering approach of [10], builds a probabilistic model based on a set of m boolean, manually defined features used to describe the documents. The collaborative filtering matrix is modified, with $\mathcal{M}_{i,j}$ reporting the probability for the i-th user (job seeker or recruiter) to select a document, conditionally to the j-th feature being satisfied by this document.

3 The corpus

The presented study is based on the corpus provided by the Qapa company, which includes 4 million job announcements, 5 million CVs, and 8 billions clicks of job seekers applying on

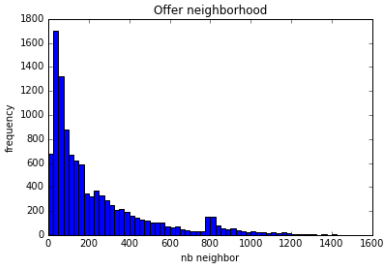


Figure 1: Distribution of recruiters neighbors, where two recruiters are neighbor if at least one job seekers applied to both propositions. The bump after the 800 mark is explained as a job seeker applied to 767 jobs; each of these 767 recruiters has at least 766 neighbors.

job announcements over the 2012-2016 period. A 2-month extract thereof has been used in the experiments, including n=143,000 job announcements and m=65,000 CVs, with $\mathcal{M}_{i,j} = 1$ iff the j-th seeker clicked on the i-th job announcement. The Qapa company specifically targets temporary contracts and low-wage jobs. The available textual information is accessible using web browsers and smartphones, often resulting in very short CVs (2 sentences, possibly a-grammatical).

A first filter, removing all CVs and all job announcements but those with at least 5 clicks, retains $n = 11,000$ recruiters and $m = 7,500$ CV, where matrix $\mathcal{M}$ reports 84,000 clicks.

The structure of the collaborative filtering matrix $\mathcal{M}$ is analyzed together with the contents of the CV and job announcement documents.

3.1 Exploratory analysis

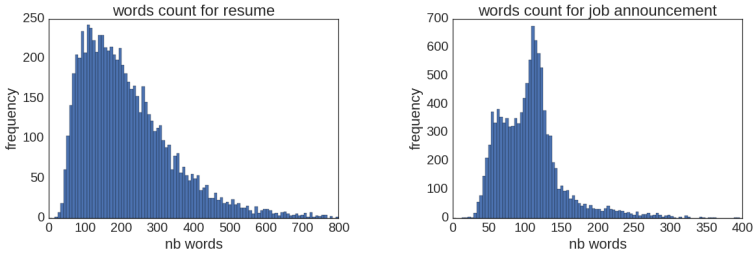
Matrix $\mathcal{M}$ is first characterized from the median number of clicks, that is 7 for the job seekers and 6 for recruiters. Let two recruiters be defined as neighbors if they have been clicked by (at least) one same job seeker. The neighbors heavy-tailed distribution is displayed in Fig 1. The median (resp. average) number of neighbors is 139 (resp. 210), which is explained as a few job seekers click on a huge number of announcements.

The sparsity of $\mathcal{M}$ is 0.1% (compared to 1% in e.g., Netflix). Complementary experiments suggest that 0 corresponds to a "don't like" much more often than in standard recommendation setting, where 0 more often corresponds to "don't know".

After tokenization and stop words removal, job announcements and CVs contain 9000 unique words. Terms made of 2 and 3 consecutive words have also been considered, resulting in a 30,000-sized vocabulary; only the 7,000 most frequent ones are used in the experiments.

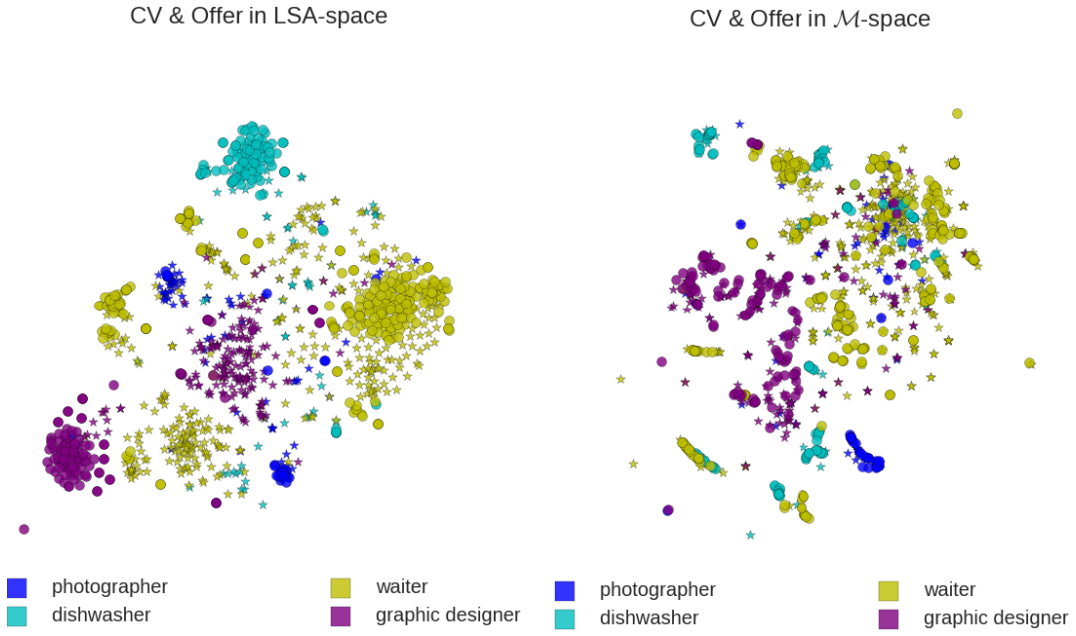
Fig. 2 reports the distribution of the lengths of CVs (left) and job announcements (right). Note that the support of the job announcement distribution is more compact than for CVs. As said, the CVs acquired through Web or mobile phone are broadly diversified and often informal, e.g., "physical reception; management of communication lines; appointment scheduling; maintenance of medical supplies"; "The mission that I had in the boat that was storing the car and cleaning the deck"; "team leader, management, HACCP norm, Chef de cuisine".[2]

[2]In contrast, a typical recruiter post would read: "Administrative assistant : Company of industrial cleaning, sweeping and sanitation. In connection with the accounting department, you will mainly billing. You have a good computer skills (internal software on which you will be trained). You have accounting concepts.".



(a) CVs (b) job announcements

Figure 2: Distribution of the document lengths.



(a) Representation of documents (LSA) (b) Representation of interactions (SVD)

Figure 3: Visual representation of job announcements (circles) and CVs (stars). Left: the textual information is processed using LSA (dimension 1,200, projected in 2D using t-SNE [16]); for some categories such as graphic designers and waiters, the job announcements and CVs form distinct clusters, suggesting that job seekers and recruited do not use the same vocabulary. Right: the interaction information is processed using SVD (dimension 500, projected in 2D using t-SNE); job seekers and recruiters are localised in same clusters; the fact that job seekers might click on different job categories (e.g. waiters and dishwashers) is also visible. Figure better seen in color.

3.2 Job seekers and recruiters do not speak the same language

A most interesting finding is that, although job seekers and recruiters undoubtedly understand each other, CVs and job announcements are not expressed in the same language, at least for some job categories. This claim is supported by conducting the following experiment on the textual

description (CVs and job announcements using the 7,000 vocabulary) on the one hand and the interaction matrix $\mathcal{M}$ on the other hand. The singular value decomposition (in dimension 1,200 for the textual description and in dimension 500 for $\mathcal{M}$) yields a vectorial representation of the job seekers and recruiters. This representation, together with the Euclidean metrics, is used to map the job seekers and the recruiters on the 2D plane with t-SNE [16].

Fig. 3 represents four job categories: photographers, dish washers, waiters and graphic designers, in textual space (left) and in interaction space (right). Photographers form a dense cluster in both textual and interaction spaces. Dishwashers form a single cluster in the textual space, and several clusters in the interaction space. For waiters, and even more so for graphic designers, two very distinct clusters are observed in the textual space: one for job seekers (legend star), and one for recruiters (legend circle).

The separation of the job seeker/recruiter clusters in the textual space is explained as people are indeed using different languages, e.g. "You take in charge the physical reception of the patients, the management of the planning and the patient records" (job announcement) versus "Secretary accountant; Capture and storage of documents" (CV).

The separation of the job seeker/recruiter clusters disappears when considering the interaction space (Fig. 3, right): each job category involves several clusters, but each cluster mixes job seekers and recruiters. The fragmentation of e.g., the graphic designer clusters in the interaction space is explained as these clusters capture some information such as the geographic localization, which is hardly captured in the textual information.

3.3 Discussion

The difference of language between the two populations might give some additional empirical support to e.g. the frictional unemployment theory, emphasizing the cost of acquiring and processing the relevant information on the job market. It might also explain the difficulties of NLP-based matching of job announcements and CVs[3], and the mandatory use of external resources, such as ontologies. Another approach will be proposed in this paper, using the interaction-based representation to achieve job-CVs matching.

4 Overview of MAJORE

Aimed at recommending job seekers to recruiters, MAJORE involves two modes respectively referred to as collaborative filtering (CF) and cold-start (CS) modes. Its input includes the collaborative matrix $\mathcal{M}$ recording the clicks of known job seekers applying on known job announcements, and the textual description of CVs and job announcements. For simplicity, it will be assumed in the remainder of the paper that a recruiter is interested in a single job announcement; the two terms will be used interchangeably.

In CS mode, the recruiter is only known from the job announcement document. In CF mode, the recruiter description also includes the job seekers having clicked on the job announcement, known from matrix $\mathcal{M}$.

4.1 MAJORE Architecture

MAJORE involves three modules. The first module computes standard vectorial representations, mapping each recruiter/seeker onto $\mathbb{R}^d$. The second module builds a metrics on $\mathbb{R}^d$, either

[3] The fact that merely matching the key words in job announcements and CVs yields poor results will be evidenced in section 6 (methods tf-idf$_m$ and LSA_m).

directly, or using matrix $\mathcal{M}$. The third module, implemented as a neural net, trains a specific representation for the job matching task. This representation, hybridizing Collaborative Filtering and Locally Linear Embedding [20] and called CLE for Collaborative Local Embedding, is the core of the MAJORE contribution.

In the following, a recruiter (respectively a job seeker) is denoted x (resp. y).

4.2 Standard representations and metrics

Four standard representations are defined. The former two are defined from the textual information; the latter two are defined from matrix $\mathcal{M}$ and hence only available for known job seekers and recruiters:

- Primary representation. The primary representation of a recruiter (resp. a job seeker) is the tf-idf representation of the job announcement (resp. CV) with dimension 7,000 (vocabulary size).
- LSA representation. The LSA representation is obtained through singular value decomposition of the primary representation, with dimension (SVD rank) $k = 1,200$ in the experiments.
- Collaborative primary representation. The interaction matrix $\mathcal{M}$ is normalized by column and then by row, forming the matrix $\widehat{\mathcal{M}}$. The collaborative primary representation of the i-th recruiter (respectively, j-th seeker) is given by the i-th row (resp., j-th column) of matrix $\widehat{\mathcal{M}}$.
- Collaborative SVD representation. This representation is obtained through singular value decomposition of $\widehat{\mathcal{M}}$, with dimension ℓ set to 500 in the experiments.

Two similarities are defined on the top of the above representations. The former one, noted sim is defined by the scalar product of two vectors. The latter similarity, defined on (recruiter, job seeker) pairs and noted sim_c, is obtained by convolution of the former similarity with matrix $\mathcal{M}$. Thus sim_c gives a matching score between a CV and a job.

$$sim_c(x, y) \propto \sum_{x'} sim(x, x')\mathcal{M}_{x',y} \tag{3}$$

4.3 The CLE representation

As will be seen in section 6, the job matching performances are excellent in collaborative filtering mode while the baseline results in cold-start mode are poor.

It thus comes naturally to view the collaborative representations extracted from matrix $\mathcal{M}$ (either the collaborative primary or the collaborative SVD representations) as an *oracle representation*, defining a relevant metrics on the job seeker and recruiter spaces. The goal thus becomes to learn a new representation from the primary representation − the only representation available in cold-start mode − such that it preserves the good topology of the oracle representation.

The proposed approach takes inspiration from the Locally Linear Embedding (LLE) approach [20]. LLE, aimed at non-linear dimensionality reduction, expresses any point x_i in the initial representation as the weighted sum of its p nearest neighbors, where p is a hyper-parameter of the approach, such that the weights sum to 1:

$$\forall i, x_i \approx \sum_j w_{i,j} x_j \ s.t.\ w_{i,i} = 0,\ \sum_j w_{i,j} = 1 \text{ and } \sum_j I_{(w_{i,j} \neq 0)} = p$$

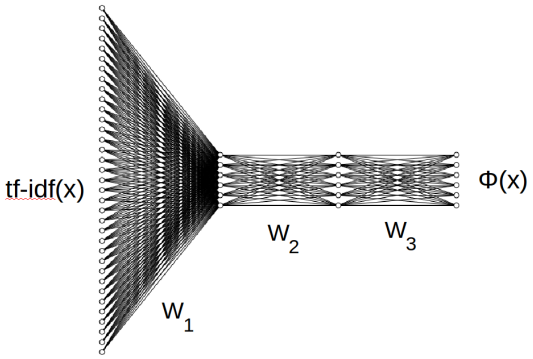


Figure 4: The CLE architecture: the number of input neurons is the vocabulary size (7,000); the number of neurons in layers 2 and 3 is the dimension k of the LSA representation ($k = 1,200$).

with I_e the indicator function of event e (thus ensuring that the numbers of neighbors is equal to p). Dimensionality reduction is thereafter achieved by looking for points z_i preserving the same relations as the x_is, i.e. such that

$$\forall i, z_i \approx \sum_j w_{i,j} z_j$$

Interestingly, the LLE mapping (mapping each x_i onto $\phi(x_i)$) is invariant by translation, homothety and rotation of the x_is.

The LLE principle is adapted to find a mapping ϕ from the initial representation $\mathbb{R}^D$ onto $\mathbb{R}^d$ such that

$$\phi^* = \arg\min \left\{ \sum_i \left(\phi(x_i) - \sum_j w_{i,j} \phi(x_j) \right)^2 - \alpha R(\phi) \right\} \tag{4}$$

where regularizing term $R(\phi)$, set to the variance of the $\phi(x)$s, is meant to avoid the trivial solution $\phi(x) = 0$.

Algorithmically, the CLE representation is learned by training a neural net to minimize Eq. 4. The mapping from the input to the first hidden layer is initialized using the LSA decomposition; the following layers are initialized to random matrix (Fig. 4).

4.4 The MAJORE options

Finally, MAJORE involves 8 options, summarized in Table 1.

Metrics	Representation				
	tf-idf	LSA	$\widehat{\mathcal{M}}$	SVD	CLE
scalar product	tf-idf$_m$	LSA_m		SVD_m	
convolution with $\mathcal{M}$ (Eq. 3)	tf-idf$_u$	LSA_u	$\widehat{\mathcal{M}}_u$	SVD_u	CLE$_u$

Table 1: The MAJORE options

Subscript m refers to model-based options, only using the textual information (*tf-idf*$_m$ computes the scalar product of job announcement and CV words; LSA_m achieves the dimensionality

reduction of the documents) or the interaction matrix (SVD_m is the dimensionality reduction of interaction matrix $\mathcal{M}$). All model-based options yield a vector representation for jobs and CVs, enabling to reconstruct $\mathcal{M}$ (Eq. 2). Subscript u refers to user-based options, indicating that the considered representation (textual, latent or neural) is used in convolution with matrix $\mathcal{M}$ (more precisely, $\widehat{\mathcal{M}}$) to define a finer-grained metrics (Eq. 3). Note that the user-based option only requires the textual description of the recruiter.

5 Experimental Setting and Goals

The goal of the experiments is threefold. The first one consists of assessing the quality of the data, already discussed in section 3. The second one concerns the performance of the MAJORE algorithm in collaborative filtering mode.The third goal concerns the MAJORE performance in cold-start mode, which is the most relevant one from the application viewpoint.
In CF mode, the experimental setting is as follows:

- For each job announcement, one click is randomly removed, defining a lesioned matrix $\mathcal{M}'$.
- MAJORE.CF is applied on the representation made of all documents and $\mathcal{M}'$;
- For each job announcement, the CVs are ranked by decreasing matching score;
- The rank $r(j)$ of the removed CV is computed;
- The first performance indicator noted $Recall_{@200}$(CF) reports the percentage of job announcements (averaged over 20 independent drawings of $\mathcal{M}'$) where $r(j)$ is less than 200.
- The second performance indicator noted AUC_{200}(CF) reports the area under the recall curve truncated at 200, likewise averaged over 20 independent drawings of $\mathcal{M}'$.

In CS mode, the experimental setting is as follows:

- Recruiters are equi-partitioned in 20 folds;
- Matrix $\mathcal{M}_{-i}$, involving all recruiters except those in the i-th subset, is considered;
- MAJORE.CS is applied on the representation made of all documents and $\mathcal{M}_{-i}$;
- For each job announcement in the i-th subset, the CVs are ranked by decreasing matching score;
- The percentage of true CVs with rank less than 200 is computed, averaged over the 20 folds and over 20 different partitions of the folds; it yields the indicator noted $Recall_{@200}$(CS).

The MAJORE hyper-parameters are summarized in Table 2. The computation times are measured on IntelCore i7-4800MQ 2.70GHz $\times$ 8 with Ubuntu 14.04. LSA with $k = 1,200$ requires less than 2 minute computational time.

The CLE neural net (with 7000-1200-1200-1200 architecture and tanh neurons), is trained and converges after a couple of epochs (with 11,000 examples per epoch) with computational time 30 mn. On-going research is investigating the impact of the number of layers and the type of neuron in the neural net.

6 Empirical Validation of MAJORE

This section presents and discusses the results obtained for the empirical validation of the MAJORE algorithm in collaborative filtering (section 6.1) and cold start (section 6.2) modes.

Hyper parameters	Range	Model
Vocabulary size	7,000	tf-idf, LSA, CLE
Latent dimension ℓ	500	SVD
Latent dimension k	1,200	LSA
Hidden layer size	1,200	CLE
Number of hidden layers	2	CLE
Nonlinearity function	*tanh*	CLE
α	1	CLE
Learning rate	20	CLE
Batch size	100	CLE

Table 2: The MAJORE hyper parameters

Score	$\widehat{\mathcal{M}}_u$	SVD_u	SVD_m	LSA_u	LSA_m
AUC_{200}(CF)	**.80 ± .01**	.79 ± .01	.68 ± .01	.57 ± .02	.31 ± .02
$Recall_{@200}$(CF)	**.93 ± .01**	.93 ± .01	.82 ± .01	.74 ± .02	.45 ± .02

Table 3: MAJORE Collaborative Filtering Performance: percentage of cases where the relevant recommendation is in the top-200, and Area Under the Recall curve (averaged over 20 runs). The dimension of the LSA (resp. SVD) decomposition is 1200 (resp. 500).

6.1 Collaborative Filtering Performances

The very poor performance of the word-only based approach tf-idf$_m$ has been omitted (see section 6.2). The counter-performance of the baseline metrics LSA_m on the top of the LSA representation is blamed on the poor overlap of the job announcement and CV vocabularies, discussed in section 3. The limitations of the LSA representation can be alleviated using the convolutional metrics (legend LSA_u), with an improvement of circa 25%.

Two main lessons are learned. On the one hand, the relevant information is captured by matrix $\mathcal{M}$, reflecting the interaction history: SVD_m, using $\mathcal{M}$ singular value decomposition with a baseline metrics yields more than twice better results than using the textual information with LSA_m. Still, the convolutional metrics SVD_u improves by more than 10% on the baseline metrics SVD_m; a tentative interpretation for this improvement is that the convolutional metrics compensates for the sparsity of $\mathcal{M}$. Finally, the best results are obtained by $\widehat{\mathcal{M}}_u$, combining the convolutional metrics with $\mathcal{M}$ (though the difference between $\widehat{\mathcal{M}}_u$ and SVD_u vanishes as the rank of the singular value decomposition increases).

The complementarity of the textual and collaborative information is shown in Fig. 5, displaying a few hundred CVs in the 2D plane defined from i) its rank according to LSA_u (x axis) and ii) its rank according to $\widehat{\mathcal{M}}_u$ (y axis), the ideal result being located at (1,1). The vast majority of points have a very low rank according to $\widehat{\mathcal{M}}_u$. A few others, with high rank after $\widehat{\mathcal{M}}_u$, have a low rank according to LSA_u; however the detailed inspection of these cases did not reveal any general pattern up to now.

The fact that 80% of the relevant resumes are found in the top-200 recommended resumes is very satisfactory from the application viewpoint, as each recruiter (resp. job seeker) commonly examines several hundred CVs (resp. job announcements). This good result confirms the quality and representativity of the data. As already said however, the collaborative filtering setting does not correspond to the general setting of the job matching application, that mostly considers

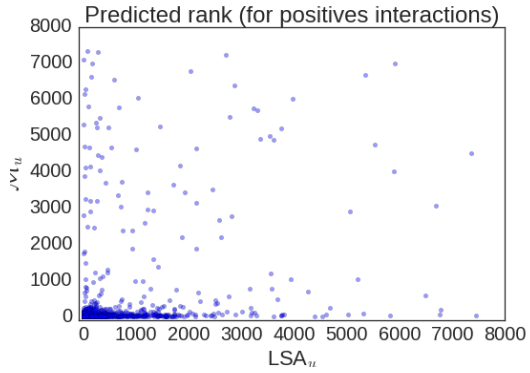


Figure 5: Respective performance of LSA_u (x axis) and $\widehat{\mathcal{M}}_u$ (y axis), reporting the rank respectively attributed to 2,000 CVs (the lower the better). While $\widehat{\mathcal{M}}_u$ outperforms LSA_u on average, a small but non negligible minority of CVs is well ranked by LSA_u and badly ranked by $\widehat{\mathcal{M}}_u$.

Score	tf-idf$_m$	tf-idf$_u$	LSA_m	LSA_u	Cle$_u$
AUC_{200}(Cs)	.32 ± .01	.35 ± .01	.29 ± .01	.51 ± .01	**.55** ± .01
$Recall_{@200}$(Cs)	.48 ± .01	.51 ± .01	.42 ± .02	.68 ± .01	**.74** ± .01

Table 4: Majore Cold Start Performances: percentage of relevant recommendations in the top-200, averaged over 20 folds and 20 runs.

new job announcements and CVs.

6.2 Cold Start Performances

Table 4 describes the cold-start performances averaged on 20 runs; representative recall curves are depicted in Fig 6.

The shortcomings of the textual representation are confirmed by the poor performance of the baseline metrics in tf-idf and LSA representation. The use of the convolutional metrics significantly improves the result, for the raw tf-idf representation and even more so for the LSA representation. The best performances are observed for the Cle representation, and Fig. 6 confirms that Cle consistently outperforms all other options whatever the threshold value.

The slight differences observed for LSA_m and LSA_u between the Cf and Cs modes are explained as i) the LSA representation is not extracted from the same data (all job announcements are considered in Cf); ii) the performance indicator does not measure the same information: in Cf mode, one examines whether *one* missing click is ranked in the top 200 on average; in Cs mode, one examines the fraction of missing clicks (over all job announcements) ranked in the top 200; the last indicator is biased toward job seekers with more clicks, who are harder to rank due to their diversity.

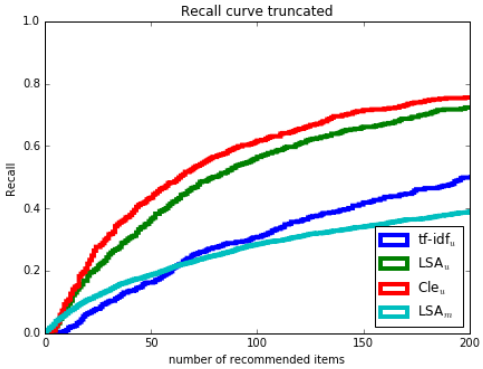


Figure 6: MAJORE Cold Start Recall curves: $x \rightarrow$ percentage of relevant recommendations with rank $< x$

7 Discussion and Perspectives

A first contribution of the paper is to shed some empirical light on the difficulties met by e-recruitment, by confronting the textual information (what job seekers and recruiters say) and the behavioral information (on which announcements do they click). As could have been expected, what people do might be rather different from what they say. People easily overcome this difference by "reading between the lines" to see whether a given document is relevant to their search. A different story is whether an automatic system can be educated to read between the lines in the same way.

Answering this question leads to further questions. Firstly, how much data is required in order to do so? Secondly, how fast is the e-recruitment domain evolving? Typically, the availability of the e-recruitment platform on mobile phones already had a very significant impact on the information content of the CVs.

A promising approach is presented in this paper, using a deep neural network to emulate the metric properties of the oracle, collaborative filtering-based, representation.

More complex architectures (e.g., handling the geographic information; considering domain adaptation among the job seeker and the recruiter spaces [12]) will be considered in further work. A main challenge will be to adapt to the evolution of the actual user behaviors, responding to the evolution of the job matching platform.

Acknowledgments

The first author's PhD is funded by a grant of the Lidex *Institut de la Société Numérique* from the Paris-Saclay University. All authors warmly thank S. Delestre who made this study possible.

References

[1] Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, and Raghu Ramakrishnan. Content recommendation on web portals. *Communications of the ACM*, 56(6):92–101, 2013.

[2] Shaha T Al-Otaibi and Mourad Ykhlef. A survey of job recommender systems. *International Journal of the Physical Sciences*, 7(29):5127–5142, 2012.

[3] Y. Bachrach. Human judgments in hiring decisions based on online social network profiles. In *Data Science and Advanced Analytics*, pages 1–10, 2015.

[4] Y. Bengio, A. C. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013.

[5] J. Bennett and S. Lanning. The netflix prize. In *Proc Int. Conf. on Knowledge Discovery and Data Mining (KDD) Cup and Workshop*, page 35, 2007.

[6] Jacob Bollinger, David Hardtke, and Ben Martin. Using social data for resume job matching. In *Proceedings of the 2012 Workshop on Data-driven User Behavioral Modelling and Mining from Social Media*, DUBMMSM '12, pages 27–30, 2012.

[7] Fernando Diaz, Donald Metzler, and Sihem Amer-Yahia. Relevance and ranking in online dating systems. In *Proc. Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 66–73, 2010.

[8] Richard Doherty. Getting social with recruitment. *Strategic HR Review*, 9(6):11–15, 2010.

[9] E. Faliagka, K. Ramantas, A. Tsakalidis, and G. Tzimas. Application of machine learning algorithms to an online recruitment system. In *Proc. International Conference on Internet and Web Applications and Services*, 2012.

[10] Frank Färber, Tim Weitzel, and Tobias Keim. An automated recommendation approach to selection in personnel recruitment. *AMCIS 2003 Proceedings*, page 302, 2003.

[11] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Commun. ACM*, 30(11):964–971, November 1987.

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, to appear in 2016.

[13] Rémy Kessler, Nicolas Béchet, Mathieu Roche, Juan-Manuel Torres-Moreno, and Marc El-Bèze. A hybrid approach to managing job offers and candidates. *Inf. Process. Manage.*, 48(6).

[14] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[15] V. Senthil Kumaran and A. Sankar. Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping expert. *Int. J. Metadata Semant. Ontologies*, 8(1):56–64, 2013.

[16] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[17] Jochen Malinowski, Tobias Keim, Oliver Wendt, and Tim Weitzel. Matching people and jobs: A bilateral recommendation approach. In *Proc. Annual Hawaii International Conference on System Science*, volume 6, pages 137c–137c. IEEE, 2006.

[18] T Mine, T Kakuta, and A Ono. Reciprocal recommendation for job matching with bidirectional feedback. In *Advanced Applied Informatics (IIAIAAI)*, 2013.

[19] Ioannis Paparrizos, B. Barla Cambazoglu, and Aristides Gionis. Machine learned job recommendation. In *Proc. ACM Conference on Recommender Systems*, RecSys '11, 2011.

[20] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[21] Tuukka Ruotsalo, Giulio Jacucci, Petri Myllymäki, and Samuel Kaski. Interactive intent modeling: Information discovery beyond search. *Commun. ACM*, 58(1):86–92, 2014.

[22] G. Strang. *Linear Algebra and its Applications*. Academic Press, New York, 1980.

[23] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.

[24] Xing Yi, James Allan, and W. Bruce Croft. Matching resumes and jobs based on relevance models. In *Proc. SIGIR Conference on Research and Development in Information Retrieval*, pages 809–810. ACM, 2007.