Job Recommendation From Semantic Similarity of LinkedIn Users' Skills

Conference Paper · January 2016

DOI: 10.5220/0005702302700277

Giacomo Domeniconi
University of Bologna
15 PUBLICATIONS **64** CITATIONS

SEE PROFILE

Gianluca Moro
University of Bologna
80 PUBLICATIONS **606** CITATIONS

SEE PROFILE

Andrea Pagliarani
University of Bologna
6 PUBLICATIONS **13** CITATIONS

SEE PROFILE

Roberto Pasolini
University of Bologna
11 PUBLICATIONS **54** CITATIONS

SEE PROFILE

Job Recommendation From Semantic Similarity of LinkedIn Users' Skills

Giacomo Domeniconi, Gianluca Moro, Andrea Pagliarani, Karin Pasini and Roberto Pasolini
DISI, Università degli Studi di Bologna, Via Venezia 52, Cesena, Italy
{giacomo.domeniconi, gianluca.moro, andrea.pagliarani12, roberto.pasolini}@unibo.it
karin.pasini@studio.unibo.it

Keywords: Job Seeking, Hierarchical Clustering, Latent Semantic Analysis, Recruiting, Recruitment

Abstract: Until recently *job seeking* has been a tricky, tedious and time consuming process, because people looking for a new position had to collect information from many different sources. *Job recommendation systems* have been proposed in order to automate and simplify this task, also increasing its effectiveness. However, current approaches rely on scarce manually collected data that often do not completely reveal people skills. Our work aims to find out relationships between jobs and people skills making use of data from LinkedIn users' public profiles. Semantic associations arise by applying *Latent Semantic Analysis* (LSA). We use the mined semantics to obtain a hierarchical clustering of job positions and to build a job recommendation system. The outcome proves the effectiveness of our method in recommending job positions. Anyway, we argue that our approach is definitely general, because the extracted semantics could be worthy not only for job recommendation systems but also for recruiting systems. Furthermore, we point out that both the hierarchical clustering and the recommendation system do not require parameters to be tuned.

1 INTRODUCTION

Job hunting (or job seeking) refers to the process people looking for a job perform in order to find it. Differently, finding out the right employee is a key aspect for enterprises, which continuously have to recruit according to their current needs. Both tasks are sides of the same general problem, namely allowing communication between companies and potential applicants for the sake of establishing an employment relationship. Since each manual search is onerous and tedious, methods exist that help automating these processes, such as *job recommendation systems* (Paparrizos et al., 2011; Malinowski et al., 2006) on the one hand and *recruiting systems* (Lee, 2007; Eckhardt et al., 2008) on the other hand. The former cope with the problem of automatically finding a job which is as inherent to people skills as possible. Vice versa, the latter are used by Human Resources departments to select candidates fitting the skills enterprises are looking for. The concept of skills is crucial in both previous mentioned tasks, because it could help pointing out people capabilities even better than in the state of the art approaches, which only focus on either academic degree (Dinesh and Radhika, 2014; Chirumamilla et al., 2014) or preceding job positions (Paparrizos et al., 2011).

Nowadays, except from custom private solutions possibly built in-house, social networks like LinkedIn[1], Facebook[2] and Twitter[3] are the most used recruiting systems by enterprises, because information available in such context is proven to be useful (Flecke, 2015). (Davison et al., 2011) pointed out that LinkedIn provides more accurate information compared to Facebook because everybody in a person's network can easily contradict her assertions. This can be a reason why (Zide et al., 2014) define LinkedIn as the world's largest professional network. In addition to its reliability, LinkedIn also offers recruiter accounts aiming to support the recruiting process, so that about 94% of recruiters use it (Kasper, 2015). Instead, the same trend does not hold within social media job seekers, where only 40% makes use of this network, although members are sometimes notified of possibly interesting job offers. LinkedIn professional secrecy does not allow us a complete understanding of the techniques used to recommend job positions. Anyway, analyzing some public profiles and the rela-

[1] `www.linkedin.com`

[2] `www.facebook.com`

[3] `twitter.com`

tive recommended job positions, we notice that there are often wrongly retrieved (i.e. not interesting) offers because of homonymy. This issue could make the job seeking process less effective, more manually conducted and time consuming.

Diversely, a job recommendation system should match requests and offers of jobs by favouring the best possible fit among candidates and companies according to people skills and companies' needs.

In this paper, we introduce a job recommendation system based on *Latent Semantic Analysis* (LSA) (Dumais et al., 1988) for the support in the job seeking process, evaluating its performance through a hierarchical clustering of job positions. Clustering is useful to partition data into previously unknown groups of similar items and is applied in a large variety of contexts (Cerroni et al., 2015). Hierarchical clustering aids to build a folksonomy (a user-defined taxonomy) of jobs useful to correlate them, whereas normally only each person's job positions are available as plain text. So, the basic idea is to discover similarity between different job positions and then to find out their latent associations with people skills.

Job positions are represented as vectors of skills and mapped into a transformed space by applying *LSA* to the skill-position co-occurrence matrix. Then, a complete-linkage hierarchical clustering technique is applied to correlate the transformed job positions, using cosine similarity as inter-cluster distance measure. Instead, the job recommendation algorithm we propose aims to suggest a list of recommended jobs that fit people skills. In fact, people are represented as vectors of skills just like job positions. The algorithm starts from a skill-position matrix built with training data and expanded by applying *LSA*. Afterwards, cosine similarity between people and positions in the skill-position matrix is computed for each test instance. Thus, since the algorithm basically outputs how much jobs fit people skills, an ordered list of recommended job positions can be built according to their similarity with each person's skills.

To evaluate our method, we take LinkedIn as reference scenario because of its widespread use, crawling real public profiles from which we can easily infer information about people skills and current job positions. We assume that current job position is the one fitting best the skill set of each person (i.e. the label of each test vector), although we are aware that this criteria is only partly correct, because somebody's job might not fit her skills. Then, we perform classification assigning the most likely *k* positions to each test vector; finally, we test performance by computing the maximum recall within the *k* suggested positions, exploiting the previously built hierarchy. To the best of our knowledge, there are no works about job recommendation exploiting either LinkedIn or other social networks. Our approach is therefore not directly comparable with the state of the art, because we focus on large scale data in terms of both job positions and skills. Moreover, differently from other approaches (Chi, 1999; Malinowski et al., 2006; Paparrizos et al., 2011), we argue that our method does not require manually collecting data, because they are already available on social networks. Finally, it can be noted that neither the hierarchical clustering of positions nor the job recommendation algorithm require parameters to be tuned. This makes our approach easy to use and profitable in real scenarios.

The rest of the paper is organised as follows. Section 2 outlines the state of the art approaches about recruiting and job recommendation systems. Section 3 introduces our methods for position clustering and job recommendation. Section 4 discusses the performed experiments. Finally, section 5 summarises results and points out possible future works.

2 RELATED WORK

The recruiting (or recruitment) process has been extensively studied in human resources field (Medsker et al., 1994; Allen and Van der Velden, 2001), giving increasing attention to the E-recruitment, namely a recruitment process based on web information (Kinder, 2000; Thompson et al., 2008) especially gathered from social networks such as LinkedIn, Facebook, Twitter, Xing (Zide et al., 2014; Flecke, 2015). The majority of these works focus on the *human resource* aspect of the recruitment process (Buettner, 2014; Paparrizos et al., 2011). Instead the job hunting problem, i.e. finding out the best job positions available in relation to users' skills and qualities, has seldom been analyzed.

Several past studies proved the helpfulness of machine learning approaches for job placement. For instance, in (Min and Emam, 2003), rules created by a decision tree are used to manage the recruitment of truck drivers. In (Chi, 1999), the authors apply principal component analysis to establish jobs that can be adequately performed by various types of disabled workers.

Some existing works (Dinesh and Radhika, 2014; Chirumamilla et al., 2014) focus on the academic degree of students, aiming to predict both their academic performance at early stage of their curricula and their placement chance, using supervised classifiers like SVMs or neural networks. (Elayidom et al., 2011) propose a decision tree based approach which helps students choosing a good branch that may fetch them placement in either rural or urban sectors.

There exist several works related to job recommendation starting from the candidate profiles (Siting et al., 2012; Paparrizos et al., 2011). (Rafter et al., 2000) propose an online Job Finder engine that uses a collaborative filtering algorithm with some user preferences. In (Malinowski et al., 2006) a bilateral people-job recommender system is proposed to match applicants to job opening profiles. Differently, (Paparrizos et al., 2011) recommend job positions to applicants based only on the job history of other employees. (Buettner, 2014) proposes a recommender system based on social network information, relying on three fit measures related to candidates. Not too different is the work by (Gupta and Garg, 2014), where candidate profile matching as well as preserving their job preferences are used.

User profiling is one of the major issues of these approaches, because retrieving, selecting and handling such data is hard. (Rubin et al., 2002) show the importance of extracurricular activities as users' skills indicator. (Paparrizos et al., 2011) define user profiles with three components: personal information, current and past professional positions, current and past educational information. Similarly, (Buettner, 2014; Gupta and Garg, 2014) use information related to companies, users, user preferences and social interactions; on the other hand, (Chi, 1999) uses a set of 41 skills. Our work deeply differs from the job recommendation approaches listed above because of the data being used. In fact, they use features related to the current and past experiences (in employment or education). Instead, we propose an approach that relies on the set of skills of a person, thus providing a prediction of the best job in relation to user's capabilities and knowledge.

According to (Zide et al., 2014), social media are seldom exploited for recruiting purposes. In their work, the authors study and find variables that recruiters assess when looking at applicants' LinkedIn profiles, such as completeness of information. As far as we know, the most similar work with respect to our proposal and used data is (Bastian et al., 2014), where a folksonomy of skills is constructed and a recommender system for skills is implemented. In particular, their goal is analyzing users' skills extracted from LinkedIn with the aim of helping users into profile skill filling. On the other hand, finding out relationships between jobs and users' skills through machine learning techniques is one of the main proposal of our work, addressing both the recruitment and the job hunting processes.

3 METHOD DESCRIPTION

We describe here the process used to perform clustering of job positions and to recommend such positions to any person given its set of skills.

We consider a set $\mathcal{U} = \{u_1, u_2, \ldots\}$ of *user profiles* (hence just *profiles*), each of them being the description of a specific person.

To each profile u is associated a set $S(u)$ of *skills*, representing the competencies which the corresponding person declares to have. The same skill may be associated to more than one profile. The set of all distinct skills is denoted by $\mathcal{S} = \{s_1, s_2, \ldots\}$.

To each profile u is also associated a current *job position* $p(u)$. The same job position may be the current one for more than one profile. We denote by $\mathcal{P} = \{p_1, p_2, \ldots\}$ the set of all distinct job positions.

3.1 Clustering of Job Positions

We are interested in obtaining a folksonomy of possible job positions (hence just *positions*) from the available data. In order to do so, we perform a hierarchical clustering of positions in $\mathcal{P}$.

A structured representation of possible positions is needed in order to measure their mutual distances. We extract a vector-based representation, where skills are used as features. Specifically, from the set $\mathcal{U}$ of known profiles, we build a $|\mathcal{S}| \times |\mathcal{P}|$ matrix $\mathbf{C}$ counting the co-occurrences between skills and positions across them. Values in $\mathbf{C}$ are computed as follows:

$$c_{i,j} = \left| u \in \mathcal{U} : s_i \in S(u) \land p_j = p(u) \right| \qquad (1)$$

In practice, $c_{i,j}$ is the number of profiles having both s_i among skills and p_j as position. Each position is then represented as a weighted mix of different skills, according to those possessed by persons employed in that position.

Intuitively, skills within the set $\mathcal{S}$ can be semantically similar to each other or even be synonyms: for example, "ms office" can be considered as strictly related to "ms word", while they are both quite unrelated to "psychology". The vector-based representation of positions would be improved by augmenting for each position (vector) the relevance of skills (features) related to those explicitly included in the mix.

This aspect has been addressed in text analysis, where correlations usually exist between terms (features) appearing throughout text documents (vectors). A well-known technique in this context is *Latent Semantic Analysis* (LSA), which employs Singular Value Decomposition (SVD) to obtain a lower-rank approximation of a term-document matrix (Dumais et al., 1988). Equivalently, we apply LSA to the skill-position matrix $\mathbf{C}$ to obtain a transformed matrix $\mathbf{C}'$.

We first decompose $\mathbf{C}$ into three factors.

$$\mathbf{C} = \mathbf{U} \cdot \Sigma \cdot \mathbf{V}^T \tag{2}$$

$\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices with eigenvectors of $\mathbf{C}$, while Σ is a diagonal matrix with eigenvalues. These matrices define a latent vector space, where skills and positions are represented by rows of $\mathbf{U}$ and $\mathbf{V}$, while values in Σ indicate the importance of each dimension of this space: lower values are supposed to represent dimensions yielding mostly noise instead of valid information. By setting all values of Σ except the r highest ones to 0 and multiplying back the three components, we obtain the transformed matrix $\mathbf{C}'$, which is a denoised approximation of $\mathbf{C}$ with rank r. For the r parameter, we choose the minimum value for which the sum of retained eigenvalues is at least 50% of the total.

The transformed matrix $\mathbf{C}'$, as the original one $\mathbf{C}$, contains for each position p_k a column vector $\mathbf{p}_k$ representing it. We evaluate the distance between two positions as the inverse of their cosine similarity.

$$d(p_a, p_b) = 1 - \cos(\mathbf{p}_a, \mathbf{p}_b) = 1 - \frac{\mathbf{p}_a \cdot \mathbf{p}_b}{||\mathbf{p}_a|| \cdot ||\mathbf{p}_b||} \tag{3}$$

The mutual distances between positions are finally given in input to a complete-linkage agglomerative clustering algorithm, which extracts a dendrogram of all positions. This dendrogram has the form of a binary tree with positions as leaves: Section 4 shows some excerpts of it obtained in our tests.

3.2 Job Recommendation

Other than extracting a consistent hierarchy of positions, the knowledge of a set of profiles can be used to infer the most advisable job positions for any other profile u_e, whose set of skills $S(u_e)$ is given.

This constitutes in practice a job recommendation system, where the best positions are suggested to any person according to her skills. While the positions of known profiles are assumed to be correct, it should be noted that there are usually multiple advisable positions corresponding to a set of skills. A recommendation system should return a set of most likely positions and all of them can be equally valid.

The recommendation method we use is simply based on representing both positions and profiles as comparable vectors and seeking for each profile the positions with the most similar vectors. Skills of the set $\mathcal{S}$ are used as features. To each profile will correspond a ranking of the known positions, of which only the first k items are usually considered.

Each profile u_e is simply represented by a binary vector $\mathbf{u}_e$, whose values are 1 in correspondence of skills known by the person and 0 elsewhere.

For what regards the vectors representing positions, we reuse results from the positions clustering method: one option is to use columns of the original skill-position co-occurrences matrix $\mathbf{C}$, another one is to use instead its low-rank approximation $\mathbf{C}'$ computed by means of LSA as described above.

In both cases, we compare the profile skills vector $\mathbf{u}_e$ to each column $\mathbf{p}_j$ by means of cosine similarity. For a given number k of positions to be recommended, we return the set $R_k(u_e)$ of k positions whose vectors are most similar to $\mathbf{u}_e$: these constitute the positions recommended for u_e.

4 EXPERIMENTS

The methodology described above to extract a hierarchy of job positions and to recommend them has been tested on a set of data extracted from LinkedIn. Operations have been carried out by software based both on the Java platform and on the open source R environment for statistical analysis.

4.1 Dataset Composition

The benchmark dataset we used has been extracted from publicly accessible LinkedIn profiles of users from Italy: for each one we considered the set of skills declared by its owner and the current job position.

Both skills and positions are specified by users as free text: many of them are present in multiple instances across profiles, but the majority of skills are only present in few or single profiles, due e.g. to typos or uncommon names. Another issue is the use of different languages across the dataset: many users filled in their profile in Italian due to being their native language, whereas many others used English to target a wider audience. Due to these aspects, the same actual skill or position can be found multiple times with different names.

We performed some preprocessing operations to obtain two disjoint groups of profiles suitable as training and test sets: the former is used to compute similarities between skills and positions and to build the hierarchy, the latter is used instead to evaluate the accuracy of the recommendation method.

Our final dataset is composed of 42,056 profiles for training and 30,639 for test, with 6,985 unique skills, 2,241 distinct positions and at least 3 skills for each profile. Distribution of both skills and positions

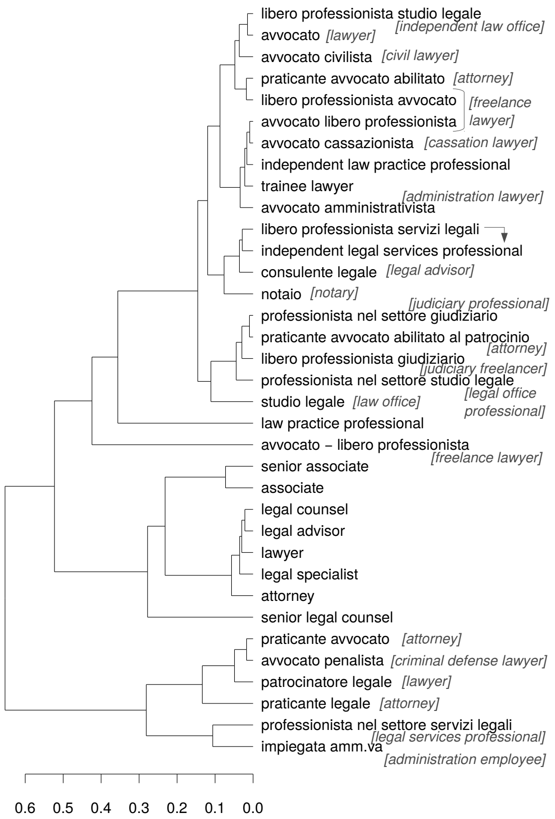
libero professionista studio legale
[independent law office]
avvocato [lawyer]
avvocato civilista [civil lawyer]
praticante avvocato abilitato [attorney]
libero professionista avvocato
[freelance lawyer]
avvocato libero professionista
avvocato cassazionista [cassation lawyer]
independent law practice professional
trainee lawyer
[administration lawyer]
avvocato amministrativista
libero professionista servizi legali
independent legal services professional
consulente legale [legal advisor]
notaio [notary]
[judiciary professional]
professionista nel settore giudiziario
praticante avvocato abilitato al patrocinio
[attorney]
libero professionista giudiziario
[judiciary freelancer]
professionista nel settore studio legale
[legal office professional]
studio legale [law office]
law practice professional
avvocato – libero professionista
[freelance lawyer]
senior associate
associate
legal counsel
legal advisor
lawyer
legal specialist
attorney
senior legal counsel
praticante avvocato [attorney]
avvocato penalista [criminal defense lawyer]
patrocinatore legale [lawyer]
praticante legale [attorney]
professionista nel settore servizi legali
[legal services professional]
impiegata amm.va
[administration employee]
0.6
0.5
0.4
0.3
0.2
0.1
0.0

Figure 1: Cluster of *legal* job positions.

is highly skewed: for example, the most recurring position is "studente" (Italian for *student*) with 1,693 training profiles and 1,383 test ones, while there are some positions with only one representative profile.

4.2 Positions Hierarchy

We applied the first step of the methodology to infer a hierarchy of job positions from the training profiles. The goal of this part is to obtain a consistent folksonomy where similar occupations are grouped together and well separated from unrelated ones.

Due to the absence of a compatible gold standard, it is not possible to quantitatively evaluate the correctness of the inferred hierarchy. Instead, we browsed through the obtained tree to check whether the obtained clusters are meaningful. As a sample, we report in Figures 1 and 2 some clusters of the hierarchy we obtained, also showing the bottom-most binary splits between elements. As discussed above, tracked positions have both English and Italian names; we provide in the figures a translation of the latter ones for readers' convenience.

From the first sample, it can be observed that the clustering algorithm mostly succeeded in outlining groups of related job positions. As we used the co-occurrences with skills shared across profiles to infer the relatedness between positions rather than words used to express them, similar occupations are effectively grouped into the same cluster, even if expressed with different terms. For the same reason, equal positions with distinct English and Italian names are mostly successfully grouped together as well.

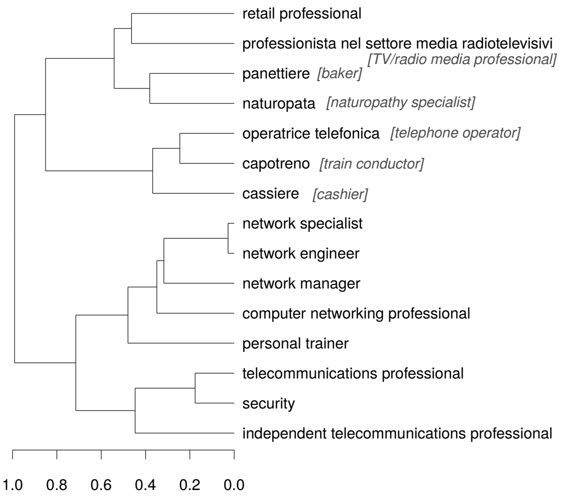
retail professional
professionista nel settore media radiotelevisivi
[TV/radio media professional]
panettiere [baker]
naturopata [naturopathy specialist]
operatrice telefonica [telephone operator]
capotreno [train conductor]
cassiere [cashier]
network specialist
network engineer
network manager
computer networking professional
personal trainer
telecommunications professional
security
independent telecommunications professional
1.0
0.8
0.6
0.4
0.2
0.0

Figure 2: Cluster of mixed job positions.

On the other hand, as the sample of Figure 2 suggests, even some unrelated positions have been possibly grouped together in the clustering. This can be due to some singularities in the co-occurrences between skills and positions in our training set. It turns out that positions associated to a sufficient number of profiles and skills have a consistent representation, whereas others that rarely occur throughout the profiles are mostly associated to unrelated skills.

For example, by looking at the sample cluster, the "personal trainer" position has been considered similar to occupations dealing with computer networks. While this appears illogical, the cause can be found in the profiles used to infer the taxonomy. Of the 9 training profiles having "personal trainer" as the current position, two declare IT and telecommunications-related skills such as "linux" and "tcp/ip". In a profile set where there are no other occupations significantly similar to "personal trainer" with sufficient occurrences, this position ends up to be grouped with unrelated ones due to some profiles declaring their peculiar skills together, thus erroneously "linking" them. Another example is the "panettiere" (Italian for *baker*) position: only two of our training profiles declare this as current employment. While one of them explicitly includes "bakery" within abilities, all the other skills of both are unrelated, mostly consisting of very generic ones, such as “teamwork” and “problem solving”, which can be equally linked to other uncommon positions.

To sum up, the obtained hierarchy successfully delineates a large number of groups of similar job positions, although with few clusters of unrelated occupations which are not sufficiently characterized in the training set. In the following we use this hierarchy to evaluate recommendations of job positions.

4.3 Results of Job Recommendation

In the second part of our experiments, we computed job recommendations for profiles of the test set, hereby denoted by $\mathcal{U}_{\text{test}}$, comparing the answers from our method with the known ones.

In our experimental evaluation, ignoring further information, we consider the current occupation of each person as the correct answer that should be given by the recommender. However, for a number of reasons, this position can’t actually be with certainty among the best possible recommendations. As discussed above, due to use of free text, a position may have many synonyms and misspelled variants indicating the same concept but considered as distinct elements of $\mathcal{P}$. A recommended job may also be strongly related to the actual one, such that it requires a very similar set of skills. Ultimately, for practical reasons, any person may be practicing a job which is notably unrelated to his or her skills. All these aspects introduce some outliers and potential errors in both training and test data, which can be detrimental for quantitative evaluations of accuracy.

The algorithm can output any number k of most recommended positions: the known position of any test profile could either be among them or not. We want to evaluate for different values of k how much frequently the recommender hits the actual positions of test profiles. For all values of k ranging from 1 to 50, we evaluated the *recall@k*, i.e. the ratio of test profiles w.r.t. their total for which the known position is among the top k recommendations.

$$\text{R@}k = \frac{|u \in \mathcal{U}_{\text{test}} : p(u) \in R_k(u)|}{|\mathcal{U}_{\text{test}}|} \tag{4}$$

As discussed above, a position given by the method for a profile may actually be a good recommendation even if different from the known one for that profile. Specifically, positions that are similar to the target one are usually equally good recommendations. We can leverage the previously computed hierarchical clustering of positions to evaluate how much a recommended position is close to the actual one. To this extent, we use the *hierarchical recall* (hR) measure proposed in (Silla and Freitas, 2011): given an actual position p_a and a single recommendation p_r, the hR is the ratio between the depth (i.e. the distance from the root, denoted here by Δ) of their deepest common ancestor (CA) and that of p_a. For k recommendations for the same profile, the maximum hR between them is considered; for the whole test set of profiles, the mean of these results is computed.

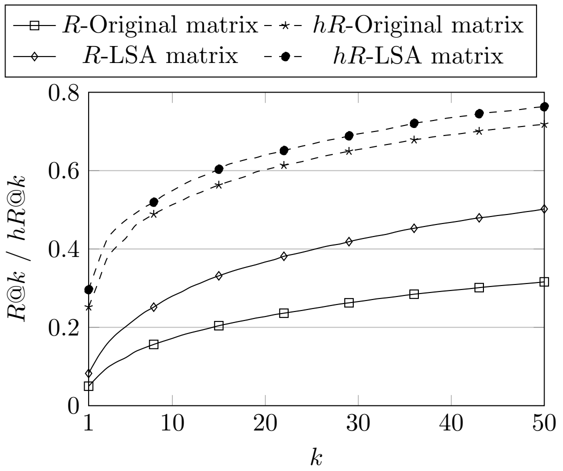

Figure 3: Trends of standard (solid lines) and hierarchical (dashed lines) recall for all values of k from 1 to 50.

Table 1: Standard and hierarchical recall for some k values.

	Original matrix		LSA matrix	
k	R@k	hR@k	R@k	hR@k
1	0.050	0.252	0.082	0.296
5	0.123	0.428	0.202	0.474
10	0.172	0.513	0.280	0.549
20	0.228	0.602	0.367	0.640
50	0.316	0.718	0.502	0.763

$$\text{hR@}k = \frac{1}{|\mathcal{U}_{\text{test}}|} \sum_{u \in \mathcal{U}_{\text{test}}} \max_{r \in R_k(u)} \frac{\Delta(\text{CA}(p(u), r))}{\Delta(p(u))} \tag{5}$$

Recommendations of job positions for all test profiles have been computed using both the described approaches, i.e. representing positions with either the original co-occurrences matrix $\mathbf{C}$ or its low-rank approximation $\mathbf{C}'$ obtained from LSA. In both cases, we compared recommendations with known positions to compute both standard and hierarchical recall for all values of k from 1 to 50. Table 1 reports recall values for some specific values of k, while the plot in Figure 3 summarizes all the measurements.

The comparison between results obtained with the two matrices shows that the use of LSA always appears to be beneficial for the accuracy of the recommendations, as it improves the representation of posi-

Table 2: Example test profiles with skills, known positions and recommendations. English translations of Italian position names are reported in italic.

Skills (alphabetical order)	Known position	Top 3 recommendations
blogging, e-commerce, facebook, marketing communications, marketing strategy, social media, social media marketing	responsabile customer service (*customer service manager*)	(1) marketing manager (2) sales manager (3) titolare (*owner*)
adults, mental health, psychology, psychotherapy	psychologist	(1) psicologa (*psychologist*, woman) (2) psicoterapeuta (*psychotherapist*) (3) psicologa psicoterapeuta (woman)

tions according to their statistically estimated relatedness. Considering this, we focus the rest of the analysis on the results for the LSA matrix.

Obviously, the accuracy grows as the number k of recommendations to be returned is raised, because it is more likely to hit the exact position or a very similar one. However, a smaller set of good recommendations can often be more valuable in practice than a larger one, which could more likely include improper elements. Looking at the standard recall, we see that a single recommendation for each profile exactly matches the known occupation in 8.2% of the test cases. As the number of recommendations grows, the known position is more likely to be hit: this happens in about one case every five with $k = 5$, one every four with $k = 8$ and one every two with $k = 50$.

Compare the standard recall to the hierarchical one for equal values of k, the latter is superior by a consistent gap, ranging between 21% and 27%. This suggests that in many cases where the exact known position is not within the recommendations, at least one of them is anyway very similar.

This can also be observed by manually comparing recommendations to known positions. Table 2 shows, for a couple of test profiles, both the actual known position and the recommended ones. It can be noted that, while the method fails at getting the exact occupation within the very top recommendations, these are nonetheless positions intuitively quite similar to it or even synonyms, which are in general equally valid and plausible for the given skills.

5 CONCLUSIONS AND FUTURE WORK

We presented a job recommendation system based on exploiting known co-occurrences between skills and potential job positions, which are elaborated by means of LSA to discover latent relationship between them. We also showed how the same data can be used to automatically build a folksonomy of job position by means of hierarchical clustering, in order to discover groups of related occupations.

The methods have been tested using a set of public profiles extracted from LinkedIn, naturally subject to noise and inconsistencies; we only applied a couple of trivial preprocessing steps to them. Despite this, we extracted a clustering where most of the groups are actually composed of related positions.

Concerning recommendations, a quantitative experimental evaluation trivially based on real job positions shows promising results, where in half of the cases the exact actual occupation of a person is within the top 50 recommended positions out of more than 2,000 possibilities. By leveraging the folksonomy of positions extracted above and looking at some specific cases we see that, even when the exact position name is not hit, homonyms and similar occupations are generally suggested.

Such a recommendation system can potentially aid both individuals seeking for occupations where their abilities can effectively be endorsed and recruiters which have to evaluate the best candidates for specific positions. The method has no parameter to be set apart from the number of recommendations to be returned, so it is simple and ready to use in practice.

One potential direction for further research would be to devise a method which fits even better to a recruitment system, for example by swapping the roles of profiles and positions, so that a set of recommended candidates can be obtained for a given occupation.

Another goal is to increase accuracy of recommendation, for example by testing other machine learning methods such as nearest neighbour classifiers or even exploiting the generated hierarchy. Also the vector representations of profiles, skills and positions could possibly be improved, for example by borrowing suitable weighting schemes from text categorization (Domeniconi et al., 2015).

Finally, we consider to test clustering and recommendation with more extended datasets, including more profiles and possibly further information for each, in order to improve the results for both tasks.

REFERENCES

Allen, J. and Van der Velden, R. (2001). Educational mismatches versus skill mismatches: effects on wages, job satisfaction, and on-the-job search. *Oxford economic papers*, pages 434–452.

Bastian, M., Hayes, M., Vaughan, W., Shah, S., Skomoroch, P., Kim, H., Uryasev, S., and Lloyd, C. (2014). Linkedin skills: large-scale topic extraction and inference. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 1–8. ACM.

Buettner, R. (2014). A framework for recommender systems in online social network recruiting: An interdisciplinary call to arms. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pages 1415–1424. IEEE.

Cerroni, W., Moro, G., Pasolini, R., and Ramilli, M. (2015). Decentralized detection of network attacks through P2P data clustering of SNMP data. *Computers & Security*, 52:1 – 16.

Chi, C.-F. (1999). A study on job placement for handicapped workers using job analysis data. *International Journal of Industrial Ergonomics*, 24(3):337 – 351.

Chirumamilla, V., Bhagya, S. T., Sasidhar, V., and Indira, S. (2014). Novel approach to predict student placement chance with decision tree induction. *nternational Journal of Systems and Technologies*, 7(1):78–88.

Davison, H. K., Maraist, C., and Bing, M. N. (2011). Friend or foe? the promise and pitfalls of using social networking sites for hr decisions. *Journal of Business and Psychology*, 26(2):153–159.

Dinesh, K. A. and Radhika, V. (2014). A survey on predicting student performance. *International Journal of Computer Science and Information Technologies*, 5(5):6147–9.

Domeniconi, G., Moro, G., Pasolini, R., and Sartori, C. (2015). A study on term weighting for text categorization: a novel supervised variant of tf.idf. In *Proceedings of the 4th International Conference on Data Management Technologies and Applications*.

Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S., and Harshman, R. (1988). Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–285. ACM.

Eckhardt, A., Laumer, S., and Weitzel, T. (2008). Extending the architecture for a next-generation holistic e-recruiting system. In *CONF-IRM 2008 Proceedings*, page 27.

Elayidom, S., Idikkula, S. M., and Alexander, J. (2011). A generalized data mining framework for placement chance prediction problems. *International Journal of Computer Applications (0975–8887) Volume*.

Flecke, L. K. (2015). Utilizing facebook, linkedin and xing as assistance tools for recruiters in the selection of job candidates based on the person-job fit.

Gupta, A. and Garg, D. (2014). Applying data mining techniques in job recommender system for considering candidate job preferences. In *Advances in Computing, Communications and Informatics (ICACCI), 2014 International Conference on*, pages 1458–1465. IEEE.

Kasper, K. (2015). Half of all job seekers arent in it for the long haul, jobvite job seeker nation study shows. Retrieved from: http://www.jobvite.com/press-releases/2015/half-job-seekers-arent-long-haul-jobvite-job-seeker-nation-study-shows/, 09-09-2015.

Kinder, T. (2000). The use of the internet in recruitment-case studies from west lothian, scotland. *Technovation*, 20(9):461–475.

Lee, I. (2007). An architecture for a next-generation holistic e-recruiting system. *Communications of the ACM*, 50(7):81–85.

Malinowski, J., Keim, T., Wendt, O., and Weitzel, T. (2006). Matching people and jobs: A bilateral recommendation approach. In *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on*, volume 6, pages 137c–137c. IEEE.

Medsker, G. J., Williams, L. J., and Holahan, P. J. (1994). A review of current practices for evaluating causal models in organizational behavior and human resources management research. *Journal of Management*, 20(2):439–464.

Min, H. and Emam, A. (2003). Developing the profiles of truck drivers for their successful recruitment and retention: a data mining approach. *International Journal of Physical Distribution & Logistics Management*, 33(2):149–162.

Paparrizos, I., Cambazoglu, B. B., and Gionis, A. (2011). Machine learned job recommendation. In *Proceedings of the fifth ACM Conference on Recommender Systems*, pages 325–328. ACM.

Rafter, R., Bradley, K., and Smyth, B. (2000). Personalised retrieval for online recruitment services. In *The BCS/IRSG 22nd Annual Colloquium on Information Retrieval (IRSG 2000), Cambridge, UK, 5-7 April, 2000*.

Rubin, R. S., Bommer, W. H., and Baldwin, T. T. (2002). Using extracurricular activity as an indicator of interpersonal skill: Prudent evaluation or recruiting malpractice? *Human Resource Management*, 41(4):441–454.

Silla, J. C. N. and Freitas, A. A. (2011). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72.

Siting, Z., Wenxing, H., Ning, Z., and Fan, Y. (2012). Job recommender systems: a survey. In *Computer Science & Education (ICCSE), 2012 7th International Conference on*, pages 920–924. IEEE.

Thompson, L. F., Braddy, P. W., and Wuensch, K. L. (2008). E–recruitment and the benefits of organizational web appeal. *Computers in Human Behavior*, 24(5):2384 – 2398. Including the Special Issue: Internet Empowerment.

Zide, J., Elman, B., and Shahani-Denning, C. (2014). Linkedin and recruitment: how profiles differ across occupations. *Employee Relations*, 36(5):583–604.