A Framework for Enriching Job Vacancies and Job Descriptions Through Bidirectional Matching

Sisay Adugna Chala, Fazel Ansari and Madjid Fathi
Institute of Knowledge Based System and Knowledge Management, University of Siegen, Holderlinstr. 3, Siegen, Germany

Keywords: Bidirectional Matching, Job Vacancy, Job Description, Text Mining, LSA, Latent Semantic Analysis.

Abstract: There is a huge online data about job descriptions which has been entered by job seekers and job holders that can be utilized to give insight into the current state of jobs. Employers also produce large volume of vacancy data online which can be exploited to portray the current demand of the job market. When preparing job vacancies, taking into account the information contained in job descriptions, and vice versa, the likelihood of getting the bidirectional match of a job description and a vacancy will be improved. To improve the quality of job descriptions and job vacancies, a mediating system is required that connects and supports job designers and employers, respectively. In this paper, we propose a framework of an automatic bidirectional matching system that measures the degree of semantic similarity of job descriptions provided by job-seeker, job-holder or job-designer against the vacancy provided by employer or job-agent. The system provides suggestions to improve both job descriptions and vacancies using a combination of text mining methods.

1 INTRODUCTION

So far, job seekers look for job vacancy advertisements and study the details of job requirements to decide whether it is suitable for their level of expertise which they have stipulated on their resume. Though vacancies are publicly available, due to overwhelming volume of data, job seekers are not able to easily find relevant vacancy for their skill or are unable to analyze the requirements to estimate its relevance. On the other hand, vacancies are not often prepared with desired skill sets required by employers (i.e., job provider). Rather, it is becoming customary that recruiters look for online profiles of potential employees from professional networking sites (e.g., Linkedin®) and/or via recommendations from networks such as partners and alliances (Sacchetti, 2013; Rafi and Shaikh, 2013; Hernandez, 2015; Godliman, 2009).

Besides, employees' job descriptions are prepared independently of job vacancy requirements and stored in professional networking sites or collected by job designers/analyzers who do job analysis research like WageIndicator(WageIndicator, 2015). However, resumeś of job seekers fail short of portraying the information available on those sites to cover all required skill sets in accordance with the job descriptions.

The measurement errors associated with the way people provide their job descriptions in relation to its effect on occupational coding are discussed by (Belloni et al., 2014). They emphasized on the need to improve the quality of job descriptions.

If there is a way that job vacancies are prepared by getting heuristic information from analysis of job descriptions (developed by job designers), taking into account the heuristic information from job vacancies (like what job seekers do), the likelihood of getting the best match of job description for a vacancy will be improved.

In this study, we propose the conception and realization of an automatic bidirectional matching system (Kucherov et al., 2014; Muderedzwa and Nyakwende, 2010) that measures the degree of semantic similarity of job descriptions provided by job-seeker, job-holder or job-designer against the vacancy provided by employer or job-agent. This similarity can then provide a feedback to improve job descriptions based on the requirements of vacancies. It also does feed-forward suggestions for the improvement to the preparation of accurate vacancies based on up-to-date job descriptions.

The rest of this paper is organized as follows: in Section 2, we discuss the state of the practice in text analysis techniques for the bidirectional match-

Chala, S., Ansari, F. and Fathi, M.
A Framework for Enriching Job Vacancies and Job Descriptions Through Bidirectional Matching.
In *Proceedings of the 12th International Conference on Web Information Systems and Technologies (WEBIST 2016) - Volume 2*, pages 219-226
ISBN: 978-989-758-186-1

ing system. In Section 3, we discuss the methodology which is dedicated to describing the data and the system setup where we discuss the source and type of data, data pre-processing, conceptual setup of the system and related algorithms. In Section 4, we describe the employed evaluation techniques. Finally we summarize the results and outline the future research works of the study.

2 BACKGROUND AND RELATED WORKS

Taking into account the challenge of matching, a combination of algorithms need to be employed for clustering of the textual data, measuring the similarity, matching and searching. Clustering refers to set of methods and algorithms for analysis of objects (e.g. graph, data, document, text or term) to identify related items, and organizing them into groups whose members are similar (Fortunato, 2010; Schaeffer, 2007; Biemann, 2012; Fasulo, 1999).

Proper selection of a clustering method (or algorithm) is highly related to the application context i.e. clustering of graphs, data, document, text or term. In data clustering - (Gan et al., 2007) - communities are set of points which are close to each other, with respect to a measure of distance or similarity (Fortunato, 2010). The latter is potentially adaptable to our approach (Charu and Zhai, 2012).

In the concept of text (or document and term) clustering there are approaches using hierarchical clustering or text mining methods (Charu and Zhai, 2012). The methods are focused on organizing text data based on similarity or association measure. The approaches are applied in a similar way for document-, text- and term- clustering (Klahold et al., 2014). So we used the term (word) clustering to refer to such methods. In this context, hierarchical term clustering algorithms (techniques) are detected such as single-link, complete-link, average-link, cliques, and stars (Li, 1990; Rajasekaran, 2005).

The single-link or single-linkage clustering method detects and merges unlinked pair of points in two clusters with the largest similarity (Manning et al., 2008), while complete-link clustering or complete-linkage clustering determines the similarity of the most dissimilar members of the clusters (Manning et al., 2008). In average-link or average-linkage, the average value of all the pairwise links between points (for which each is in one of the two clusters) is a measure for computing the similarity (William and Baeza-Yates, 1992).The clique clustering groups the data into cliques i.e. identifying subspaces of a high dimensional data space that allow better clustering than original space (Kochenberger et al., 2005; Gijswijt et al., 2007).

The main star algorithms are the Scatter-Gather - (Cutting et al., 1993) - and (Charikar et al., 1997). The star algorithms do not impose a fixed number of clusters as a constraint on the solution (Gil-García et al., 2003; Aslam et al., 2004). The algorithms detect the highest degree unmarked node and mark it as a star center, and construct a cluster from the star center and its associated satellite nodes (Gil-García et al., 2003; Aslam et al., 2004). Finally each node in the newly constructed cluster is marked (Gil-García et al., 2003; Aslam et al., 2004).

In addition to the clustering methods which are discussed earlier, there are a number of related works using ontology-based framework for text clustering (Hotho and Staab, 2002; Yang et al., 2008; Tar and S., 2011; Ma et al., 2012).

Furthermore, a number of studies have been conducted in the area of both fuzzy and exact matching of patterns (Hussain et al., 2013) in which they applied exact matching algorithm using two pointers (simultaneously) based on window sliding method where they tried to compare bidirectional algorithm's results with Quick Search, BM Horspool, Boyer-Moore and Turbo BM algorithms that are deemed to be efficient for character comparisons and attempts to complete processing of selected text. They used bidirectional matching algorithm that compares a given pattern from both sides, starting from right then from left, one character at a time within the text window and produced an algorithm that scans text string from both sides simultaneously against the given pattern. Its analysis shows that it takes $O(mn/2)$ time where m is the length of the given pattern and n is the length of the target text.

Another study by (TextKernel, 2015) uses the resumé text of the candidate's profile and automatically creates a search which is performed on multiple and multi-lingual sources of jobs. The search system collects and structures online jobs, can match them to a profile and helps quickly find relevant job for a job seeker.

In this study, our approach to document and term clustering is the extension of the work presented in (Klahold et al., 2014) which employs term similarity measures for text clustering elaborated in Section 3.4, and bidirectional matching that is focused on document matching as opposed to term matching (cf. Section 3.5).

Unlike its usage in (Hussain et al., 2013) for pattern matching, in this study, the concept *bidirectional matching* refers to matching terms in job description

with job vacancies and vice versa (cf. Figure 2) to produce a unified search space for documents in the same cluster. This study is also different from that of (TextKernel, 2015) in that it tries to generate a common terms database to represent job descriptions and vacancies in same cluster to maximize the likelyhood of their appearance in the suggestion list. It does not use active vacancies and user profiles as in (TextKernel, 2015) rather it uses user profiles, active and historical vacancies and standard job description. Its result at this stage is a representation of job descriptions and job vacancies which will be used as an input to job vacancy recommender in later stages.

3 BIDIRECTIONAL MATCHING SYSTEM

In this section, we describe the data source, methods of collection and pre-processing; how the system is conceptually set up; and the algorithm that is used to find the matching of job descriptions.

3.1 Data Source

Wageindicator database contains a huge volume of job descriptions and wage information that enables us to obtain data on what job seekers (or job holders) provide about their existing jobs. This database also contains data that job holders provide about their wages.

Job vacancy data is collected from online job advertising sites using web crawlers designed especially for this purpose. As these data are collected from plethora of sources with multilingual nature, they will be translated to a common language using machine translation systems.

We use data of varied type from multiple sources collected, stored and managed by WageIndicator (WageIndicator, 2015). First, there is a big database of job descriptions made by job designers for 430 job titles and an average of 10 sentences per job title that is available in 13 languages which was initially made by International Labor Organization (ILO) (WageIndicator, 2015).

Second, there is a growing database of job descriptions by job holders which includes 1,700 job titles coded according to ISCO-08 scheme (WageIndicator, 2015). The number of characters used by the jobholders varies largely, and the text is unstructured. This text is collected by means of the WageIndicator web survey, and the survey data includes also wages and other variables of interest. The survey is held in 85 countries, although most responses are from the Netherlands and Germany (WageIndicator, 2015). We also use the job descriptions available on European Skills, Competences, Qualifications and Occupations (ESCO)(EC, 2015).

Third, in relation to job advertisements, in a few countries, WageIndicator publishes vacancies from job advertisement companies that need to be crawled. The data collected from all the above sources is used to train and test the proposed bidirectional matching system.

3.2 Data Pre-processing

In the collected data, job holders enter their job description mixing it with their emotions and non-job-related information, hence the description may include phrases such as "I like my job but not my boss". For this reason, extensive data cleaning work is done before indexing the documents. All of the data in the collection is organized and subsequently converted to plain text. Then it is cleaned up from the blank lines and noisy characters (e.g., punctuations) and finally its encoding converted to UTF-8 automatically to make it ready for training of the system.

3.3 Conceptual Framework

As shown in Figure 1, the data collected by web crawling from online job vacancies is matched against the data from Wageindicator Foundation to find out which job descriptions align with which vacancies, and vice versa. This alignment is in turn stored in a database that contains pool of skill sets. The system is used to improve the quality of job vacancies and to assist job seekers (or job holders) during their entry of job descriptions.

3.4 Document Similarity Analysis and Clustering

The first step towards document similarity analysis is to build a document vector in order to represent the document as a whole. This is done through a statistical approach, in which the vector will be made from the statistically most important words contained in the document by removing stop words, i.e., words that are too common to distinguish documents from one another. The importance of words or terms is weighted according to their popularity in the data set. The outcome prioritizes the terms that are relatively rare in the data set.

Due to the fact that the data set is focused on a specific topic – occupation – we will also use vocabularies to guide the formation of the document vector as

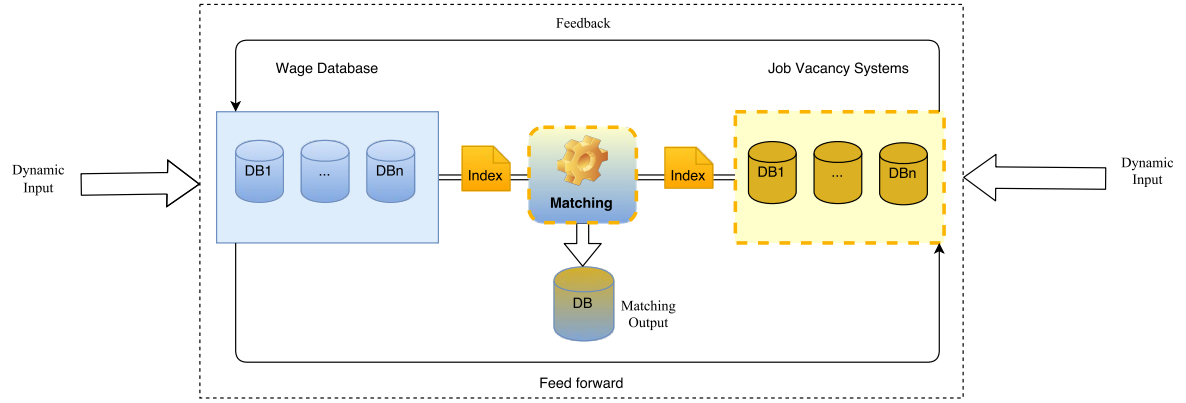

Figure 1: Conceptual Framework of the Bidirectional Matching System.

it is practical to build and maintain a suitable vocabulary for specific subject matter. In addition to the automatically extracted indexes, vocabularies from standard occupation are used in the system to improve the ability of the system to distinguish between different job descriptions.

After having extracted and stored the document vector of indexes, the similarity analysis applications work by comparing the vectors of documents using range of statistical approaches such as TF/IDF, Cosine similarity, Dice Similarity, Jaccard similarity, or Latent Semantic Analysis (LSA) techniques (Jurafsky and Martin, 2009).

We use LSA, also known as Latent Semantic Indexing (LSI), which is a technique that utilizes the concepts of vector space model and Singular Value Decomposition (SVD). It was first proposed by (Deerwester et al., 1990) to construct a weighted terms-by-documents matrix and using the matrix to represent the concepts contained in the text. This way, we build a matrix of terms-by-documents that we will use in the later stages to perform SVD on the matrix and find singular values that represent job descriptions as concepts in the document.

A simple search which only looks for existence of words fail to perform when the words are misplaced or a synonym of the words are used (Harrington, 2012; Landauer, 2007). In contrast, with LSA the synonyms represent a common concept and thus point to the same documents (Harrington, 2012; Landauer, 2007). In the context of our data which is entered mainly by job seekers' or job holders' self-assessment (Tijdens and van Klaveren, 2012), where synonym is a key issue to study, we found LSA as a viable option over the other techniques because in LSA synonyms represent a common concept.

To build the terms-by-documents matrix, M, we need to first identify the occurrences of the A unique terms minus the non-value adding terms, i.e., stop words, within a collection of B documents. In a terms-by-documents matrix, each raw represents term and each column represents document, thus forming a matrix of size AxB, where A is the number of unique terms in the dataset and B is the number of documents representing the data set. Each matrix cell, m_{ij}, represents the count of term in the corresponding document, tf_{ij}, where tf stands for term frequency (Jurafsky and Martin, 2009).

After the terms-documents matrix is built, weighting functions are applied to it to transform the data because the matrix is large and sparse. Then, the value of m_{ij} is transformed to represent the production of the relative frequency of the term in the document, l_{ij}, and the relative frequency of the term in the total document collection g_{ij}. For the local term weighting function, we use entropy method as it is proved to have performed better than the other weighting functions (Nakov, 2000).

3.5 Bidirectional Matching

The way human beings match resumes to job descriptions has been summarized into a series of steps including review of job descriptions, summarization and entry of qualifications for the descriptions' requirements (Jones, 2015).

The automated system, on the other hand, extracts text from multiple documents from various sources and splits the text into words to prepare for the matching process. This method performs a variety of different operations and text analyses related to extraction and matching of several files based on document similarities using indexes. The system is not only helpful to extract and match the specified text from multiple job descriptions but also filters out documents which do not contain a specified text. After the process of matching, the system produces the resulting matched data and stores it into the database for searching.

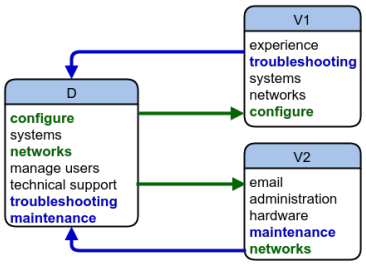

Figure 2: Bidirectional Matching: An Example.

Once the documents are represented by indexes (i.e., words or terms), we experiment to evaluate n-gram string matching using the indexes from the two sets of documents (i.e., job descriptions and job vacancies) to find out the optimal n-gram (Jurafsky and Martin, 2009).

For example, we use keys from indexes extracted from job descriptions provided by job holders to search for patterns in job vacancies provided by employers, and vice versa. Then we compare the results if they have reasonable matching. The choice of bidirectional matching is because it has been reported to perform well in pattern matching (Chatterjee and Perrizo, 2009). Moreover, it has a space complexity of O(mn/2), where m and n are the number of characters in the search space (Kucherov et al., 2014). The literature review by (Hussain et al., 2013) shows that the complexity of bidirectional matching is better than that of all other pattern matching algorithms for text processing. For instance, considering a job description for a System Administrator, one can find a number of vacancies with different job requirements. Let us take part of the job description $D \in \{D_1, D_2, ...D_n\}$ with content *"... ability to configure systems and networks, manage users, give technical support to users ..."* and two example vacancies V_1 and $V_2 \in \{V_1, V_2, ...V_m\}$ with content *"... experience in troubleshooting systems over wide area network and proven knowledge of working on virtual private networks ..."* and *"... skill in corporate email administration and hardware maintenance ..."*, respectively. It is apparent that some of the key requirements of the vacancies such as *troubleshooting* in V_1 and *maintenance* in V_2 are missing in D. Thus, the system, on the one side, provides suggestion for the job designers to enrich D by incorporating *troubleshooting* from V_1 and *maintenance* from V_2. On the other side, the system provides recommendations to enrich V_1 and V_2 by incorporating the terms *"configure"* and *"network"* from D, respectively (cf. Figure 2, cf. Algorithm 1).

Algorithm 1: Algorithm for our Bidirectional Framework.

Require: D and V are two vectors of strings of length m and n

1: **procedure** BIDIRECTIONALMATCHING($< D_1,...,D_m >, < V_1,...,V_n >$)
2: $Cluster(< D_1,...,D_k >, < V_1,...,V_j >) \leftarrow Similar(D_i, V_j)$ ▷ cluster similar job descriptions and vacancies
3: **for** $c \in Clusters$ **do**
4: **for** $D \in D_1 \ldots D_k$ **do**
5: **for** $t_d \in D$ **do**
6: **for** $v \in V_1 \ldots V_j$ **do**
7: **if** $(t_d \notin v)$ **then** $append(v, t_d)$
8: **for** $V \in V_1 \ldots V_j$ **do**
9: **for** $t_v \in v$ **do**
10: **for** $D \in D_1 \ldots D_k$ **do**
11: **if** $(t_v \notin D)$ **then** $append(D, t_v)$
12: return $< D >, < V >$

4 EVALUATION

In this section, we discuss the methods that are employed to evaluate the effectiveness of the proposed system. To evaluate the effectiveness and sensitivity of the system, a combined precision and recall measures is employed as shown in Equation 1, and Equation 2:

$$Recall = \frac{|\{RR\} \cap \{RT\}|}{|\{RR\}|} \quad (1)$$

$$Precision = \frac{|\{RR\} \cap \{RT\}|}{|\{RT\}|} \quad (2)$$

where RR stands for the number of relevant documents and RT stands for the number of retrieved documents. In this context relevant documents refer to job vacancies and job descriptions which have matching terms with the search term whereas retrieved documents refer to job descriptions and job vacancies that are returned as suggestions based on the search terms. Thus, the effectiveness of the system is measured by how many of the job descriptions and job vacancies are suggested and from them how many of them are appropriately suggested.

While cleaning the data of non-relevant characters and terms during indexing, we also include synonyms, related terms, broad or general terms for each job description. Considering the example of *System Administrator* described in Section 3.5, for instance, *Network Administrator* is also used as a synonym of *System Administrator*.

Though these decisions affect precision negatively, because synonyms may not be exact ones and the probability of retrieving irrelevant material increases, it guarantees that overlapping job descriptions will be matched. Entries must be considered in a dichotomy of either relevant or non relevant when determining recall and precision. However, data entries have varied degree of similarity in the spectrum of totally relevant to totally irrelevant.

Referring to the example of *System Administrator* (cf. Section 3.5) some of the synonyms can be marginally relevant (e.g., Network Operator) or somewhat irrelevant (e.g., System Analyst) while others may be completely relevant (e.g., System Engineer) or completely irrelevant (e.g., Social Network Analyst). To decide this, we will use probabilistic method of rated degree of effectiveness and precision (Goutte and Gaussier, 2005). The results of matching targets are then ranked according to their degree of relevance to the indexes in the representation of the source job description (cf. Figure 2).

As shown in Equation 1 and Equation 2, recall is the proportion of relevant job descriptions and vacancies that are determined to be matching to the term in question to the total relevant documents, i.e., how many of the related job descriptions and vacancies are suggested to be matching. Precision, on the other hand, is the proportion of relevant job descriptions and vacancies that are determined to be matching to the term in question to the total suggested documents, i.e., how many of the related job descriptions and vacancies are suggested to be matching and how many of the non related ones are not suggested.

5 CONCLUSION AND FUTURE WORK

This paper presents the work aimed at matching job description with job vacancies, and explains the data together with its source and type. In addition, it elaborates the selection of algorithms for i) preprocessing and representing the documents, ii) performing the matching between job descriptions and vacancies, iii) similarity analysis, and iv) evaluating the effectiveness of the results.

The resulting representation of job descriptions and job vacancies in this study will be used as input to job vacancy recommender system.

This matching not only provides accurate and up-to-date information for job designers to develop reference job descriptions such as the ones present in standard occupation databases but also supports employers or job-agents to identify crucial and cross-cutting skill sets to be stated in the requirements for a vacancy advertisements.

There are a number of areas for future work. First, in addition to the automatic evaluation, the level of improvement in user engagement, the quality of suggestions as well as user experience will be studied and performance of the system with respect to usage will be evaluated.

Second, due to the labor force mobility especially across Europe (Fischer et al., 2014) and individual-skill mismatches (Hernandez, 2015; Godliman, 2009), the scope of this study extends to i) analyze multi-lingual job descriptions and job vacancies and ii) applying it at micro level (i.e., individual level) to support job seekers to improve the quality of their CV for a particular job posting so as to include required and preferred skills.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the financial support of the Eduworks Marie Curie Initial Training Network Project (PITN-GA-2013-608311) which is part of the European Commission's 7th Framework Programme.

Finally, the authors would like to express their deepest gratitude to WageIndicator Foundation for supporting this research.

REFERENCES

Aslam, J. A., Pelekhov, E., and Rus, D. (2004). The star clustering algorithm for static and dynamic information organization. *Journal of Graph Algorithms and Applications*, vol. 8(no. 1):95–129.

Belloni, M., Brugiavini, A., Meschi, E., and Tijdens, K. G. (2014). Measurement error in occupational coding: an analysis on share data. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2539080.

Biemann, C. (2012). *Structure Discovery in Natural Language*. Springer.

Charikar, M., Chekuri, C., Feder, T., and Motwani, R. (1997). Incremental clustering and dynamic information retrieval. *Proceedings of the 29th Symposium on Theory of Computing*.

Charu, C. A. and Zhai, C. X. (2012). *Mining Text Data*. Springer.

Chatterjee, A. and Perrizo, W. (2009). Bi-directional string matching algorithm in text mining. *IADIS Information Systems Conference*. http://www.cs.ndsu.nodak.edu/∼perrizo/saturday/papers/sede09/sede09_arijit1_bidir_string_match.pdf.

Cutting, D., Karger, D., and Pedersen, J. (1993). Constant interaction-time scatter/gather browsing of very large document collections. *Proceedings of the 16th SIGIR*.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

EC (2015). Esco home european commission. https://ec.europa.eu/esco/home.

Fasulo, D. (1999). An analysis of recent works on clustering algorithms.

Fischer, G., Strauss, R., and Maly, R. (2014). Eu employment and social situation: Recent trends in the geographical mobility of workers in the eu.

Fortunato, S. (2010). Community detection in graphs. *Journal of Physics Reports 486*, pages 75–174. DOI:10.1016/j.physrep.2009.11.002.

Gan, G., Ma, C., and Wu, J. (2007). Data clustering: Theory, algorithms, and applications. *ASA-SIAM Series on Statistics and Applied Probability)*.

Gijswijt, D., Jost, V., and Queyranne, M. (2007). Clique partitioning of interval graphs with submodular costs on the cliques. *RAIRO Operations Research*, 41:275–287. DOI:10.1051/ro:2007024.

Gil-García, R. J., Badía-Contelles, J. M., and Pons-Porrata, A. (2003). Extended star clustering algorithm. *Progress in Pattern Recognition, Speech and Image Analysis*, pages 480–487.

Godliman (2009). How to manage headhunters for candidates. http://godlimanpartners.com/interface/resources/How_To_Manage_Headhunters_-_for_Candidates.

Goutte, C. and Gaussier, E. (2005). A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. *Advances in information retrieval*, pages 345–359. http://link. springer.com/chapter/10.1007/978-3-540-31865-1_25.

Harrington, P. (2012). *Machine Learning in Action*. Manning Publications.

Hernandez, J. H. (2015). Ways to make your resume perfect for a job opening. http://www.careerealism.com/resume-perfect-match-job-opening/.

Hotho, A. Maedche, A. and Staab, S. (2002). Text clustering based on good aggregations. *Künstliche Intelligenz (KI)*, 16(4):48–54.

Hussain, I., Hassan Kazmi, S. Z., Ali Khan, I., and Mehmood, R. (2013). Improved bidirectional exact pattern matching. *Internation Journal of Scientific and Engineering Research*. https://uhdspace.uhasselt.be/dspace/handle/1942/16925.

Jones, L. (2015). How to match qualifications to a job description in a resume. http://work.chron.com/match-qualifications-job-description-resume-8135.html.

Jurafsky, D. and Martin, M. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. Prentice-Hall, 2nd edition edition.

Klahold, A., Uhr, P., Ansari, F., and Fathi, M. (2014). Using word association to detect multitopic structures in text documents. *IEEE Intelligent Systems*, 29(5):40–46.

Kochenberger, G., Glover, F., Alidaee, B., and Wang, H. (2005). Clustering of microarray data via clique partitioning. *Journal of Combinatorial Optimization*, pages 77–92.

Kucherov, G., Salikhov, K., and Tsur, D. (2014). Approximate string matching using a bidirectional index. *Lecture Notes in Computer Science*.

Landauer, T. (2007). *Handbook of Latent Semantic Analysis*. University of Colorado Institute of Cognitive Science Series. Lawrence Erlbaum Associates. https://books.google.de/books?id=jgVWCuFXePEC.

Li, X. (1990). Parallel algorithms for hierarchical clustering and clustering validity. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12:1088–1092.

Ma, J., Xu, W., Sun, Y., Turban, E., Wang, S., and Liu, O. (2012). An ontology-based text-mining method to cluster proposals for research project selection. *IEEE transactions on systems, man, and cybernetics–part a: systems and humans*, 42(3):784–790.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Muderedzwa, M. and Nyakwende, E. (2010). A framework for improving the effectiveness of it in employment screening. *In Research and Development (SCOReD), 2010 IEEE Student Conference IEEE*. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5703988.

Nakov, P. (2000). Getting better results with latent semantic indexing. *Computational Intelligence: Theory and Applications*, pages 156–166. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.108.3977&rep=rep1&type=pdf#page=164.

Rafi, M. and Shaikh, M. S. (2013). An improved semantic similarity measure for document clustering based on topic maps. *Computing Research Repository*. http://arxiv.org/abs/1303.4087.

Rajasekaran, S. (2005). Efficient parallel hierarchical clustering algorithms. *IEEE Transactions on Parallel and Distributed Systems*, vol. 16(No. 6):497–502.

Sacchetti, L. (2013). The magic of headhunting: A how-to guide to hunting and closing top candidates. http://ren-network.com/wp-content/uploads/2013/05/The-Magic-of-Headhunting-A-How-to-Guide-to-Hunting-and-Closing-Top-Candidates.pdf.

Schaeffer, S. E. (2007). Survey: Graph clustering. *Journal of Computer Science Review*, 1:27–64. Doi:10.1016/j.cosrev.2007.05.001.

Tar, H. H. and S., N. T. T. (2011). Ontology-based concept weighting for text documents. *International Conference on Information Communication and Management*, 16.

TextKernel (2015). Textkernel – cv parsing, semantic search and matching software. http://www.textkernel.com/.

Tijdens, K. and van Klaveren, M. (2012). A skill mismatch for migrant workers? evidence from wageindicator survey data. *ILPC2013*. http://www.ilpc.org.uk/Portals/56/ilpc2013-paperupload/ILPC2013paper-JP__Tijdens-Klaveren-final-18.04-kt_ILPC_20130309_121708.pdf.

WageIndicator (2015). Salary checks -world wide wage comparison. http://www.wageindicator.org.

William, B. F. and Baeza-Yates, R. (1992). *Information Retrieval: Data Structures and Algorithms*. Prentice Hall.

Yang, X., Guo, D., Cao, X., and Zhou, J. (2008). Research on ontology-based text clustering. *Third International Workshop on Semantic Media Adaptation and Personalization*, pages 14–146.