2015 IEEE 27th International Conference on Tools with Artificial Intelligence

An Automatic Online Recruitment System based on Exploiting Multiple Semantic Resources and Concept-relatedness Measures

Aseel B. Kmail
Computer Science Department
The Arab American University
Palestine
aseel.kmail@aauj.edu

Mohammed Maree
Multimedia Technology Department
The Arab American University
Palestine
mohammad.maree@aauj.edu

Mohammed Belkhatir
Campus de la Doua
University of Lyon
France
mohammed.belkhatir@univ-lyon1.fr

Saadat M. Alhashmi
Department of MIS
University of Sharjah
United Arab Emirates
salhashmi@sharjah.ac.ae

***Abstract*—Due to the rapid development of job markets, traditional recruitment methods are becoming insufficient. This is because employers often receive an enormous number of applications (usually unstructured resumes) that are difficult to process and analyze manually. To address this issue, several automatic recruitment systems have been proposed. Although these systems have proved to be more effective in processing candidate resumes and matching them to their relevant job posts, they still suffer from low precision due to limitations of their underlying techniques. On the one hand, approaches based on keyword matching ignore the semantics of the job post and resume contents; and consequently a large portion of the matching results is irrelevant. On the other hand, the more recent semantics-based models are influenced by the limitations of the used semantic resources, namely the incompleteness of the knowledge captured by such resources and their limited domain coverage. In this paper, we propose an automatic online recruitment system that employs multiple semantic resources to highlight the semantic contents of resumes and job posts. Additionally, it utilizes statistical concept-relatedness measures to further enrich the highlighted contents with relevant concepts that were not initially recognized by the used semantic resources. The proposed system has been instantiated and validated in a precision-recall based empirical framework.**

Keywords- job-to-resume matchning; candidate screening; unstructured resumes; concept-relatedness measures

I. INTRODUCTION

Recruitment is considered among the most challenging functions for job portals and human resource (HR) departments [1]. This is because employers often receive a huge number of resumes – some of which are uploaded as unstructured documents in different formats such as .pdf, .doc, and .rtf [2], while others are uploaded according to specific forms prepared by employers [3-5] – that are difficult to manually process and analyze. Recently, many companies have shifted to automatic online recruitment systems [6] in an attempt to reduce the cost, time, and efforts required for screening out applicants and matching candidate resumes to their relevant job posts [7]. As reported by SAT telecom [8], the use of online recruitment has led to 44% of cost savings and reduced the time to fill a vacancy from 70 to 37 days.

Several techniques/approaches have been employed by online recruitment systems. Examples of these techniques are Boolean Retrieval [9], models based on Relevance Feedback [10], Analytic Hierarchy Process [11], Semantics-based techniques [3, 4, 6, 12-14], and Natural Language Processing (NLP) and Machine leaning based approaches [2, 15-19]. Although these techniques achieve good matching results, they are still limited by the following obstacles. First, the use of automated keyword-based techniques to match an ever increasing number of resumes (usually in the form of unstructured text) to job posts is unsatisfying since it ignores the semantic aspects of the concepts encoded in the processed documents. Second, semantics and knowledge based techniques have drawbacks associated with the limited domain coverage and semantic knowledge incompleteness problems highlighted in [20].

To overcome the abovementioned limitations, we propose an automatic online recruitment system that exploits multiple semantic resources in an attempt to highlight and capture the semantic aspects of both job posts and candidate resumes. The proposed system employs NLP pre-processing techniques to identify and extract lists of candidate concepts from job posts and resumes. In addition, it utilizes statistical concept-relatedness measures (extracted from Hiring Solved Dataset [28]) to enrich and expand the lists of candidate concepts with entities i.e. concepts that were not initially recognized by the employed semantic resources.

We summarize the major contributions of our work as follows:

- Exploiting multiple semantic resources to cooperatively capture the semantic aspects of resumes and job posts.
- Utilizing statistical concept-relatedness measures to refine the lists of candidate concepts and enrich them with relevant concepts that were not recognized by the used semantic resources.

The remainder of this paper is structured as follows. In section 2, we introduce the work related to job-to-resume

1082-3409/15 $31.00 © 2015 IEEE
DOI 10.1109/ICTAI.2015.95

matching. A high-level overview of the proposed system architecture is presented in section 3. Section 4 details the steps of implementing the proposed system. Experimental validation and evaluation of the proposed system is presented in section 5. In section 6, we discuss the conclusions drawn from the results of the experiments and highlight the future extensions of the current work.

II. Related Work

Several techniques and approaches have been proposed to construct automatic online recruitment systems [21]. In this section, we classify these techniques and discuss the major drawbacks and limitations that are associated with each technique.

A. Traditional Keyword –based Techniques

These techniques mainly depend on exact matching between keywords extracted from job posts and candidate resumes. Systems that employ such techniques suffer from low precision wherein a large portion of the returned results is irrelevant. This is because keyword-based techniques ignore the underlying semantic aspects of the terms that are extracted from both job posts and resumes [9].

B. Relevance-based Models

Relevance models are usually built from known relevant resumes to a specific job post [22]. While in Structured Relevance Models (SRM) [23] models are built from highly ranked documents. In this context, relevance models are used to compensate for vocabulary variations between resumes and job descriptions. Similar job posts are grouped by matching a candidate job description with a collection of job descriptions. After that, resumes that are relevant to those job descriptions are used to construct relevance models to capture terms that are not explicitly mentioned in job descriptions. A major problem of these approaches is their low precision when tested against large-scale real-world datasets [23].

C. Semantics-based Approaches

As stated in [13], the exploitation of semantic resources in the recruitment domain assists in using shared vocabularies to describe job descriptions and resumes. The authors of [3-5, 12] propose automatic recruitment systems that employ semantic resources that have been built based on integrated classifications and standards. In [4] the authors propose using a human resource ontology (HR-ontology) to gain uniform representation of resumes and job posts and to accomplish the semantic matching process. Another semantics-based systems is EXPERT [14] which constructs ontology documents that describe both job posts and resumes based on the concept linking approach [25], and then ontology documents of job posts are mapped to ontology documents of resumes. Although these approaches have shown better results in accomplishing the matching process, they still face significant problems concerned with the development of complete and reliable ontologies that capture up-to-date knowledge about specific domains [13, 24].

D. Machine Learning Techniques

A number of machine learning algorithms are exploited in the online recruitment domain for data analysis and information extraction [2, 15, 16, 19]. These algorithms include neural networks [15], clustering [19], decision trees [26], and support vector machines [16]. Among the systems that employ machine learning techniques is E-Gen [16]. The authors of this system propose to automate the recruitment process through classifying and analyzing unstructured job posts using vectorial and probabilistic models. In addition, Support Vector Machine (SVM) classification techniques are employed to highlight segments of job posts with appropriate topics and features. As reported in [2], the main drawback of machine learning approaches is that they produce high error rates as they rely on manually-developed training corpora.

III. High-level Overview of the Proposed System Architecture

In this section, we present a high-level overview of the proposed system architecture and discuss its main modules.

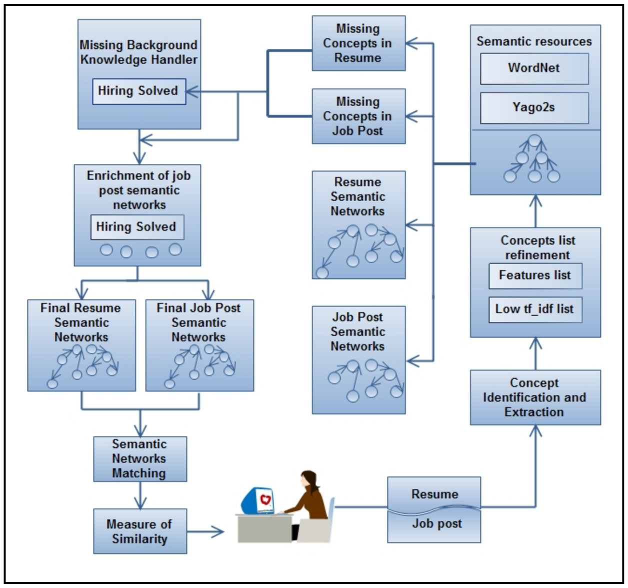
Missing Background Knowledge Handler
Hiring Solved
Missing Concepts in Resume
Missing Concepts in Job Post
Semantic resources
WordNet
Yago2s
Enrichment of job post semantic networks
Hiring Solved
Resume Semantic Networks
Concepts list refinement
Features list
Low tf_idf list
Final Resume Semantic Networks
Final Job Post Semantic Networks
Job Post Semantic Networks
Concept Identification and Extraction
Semantic Networks Matching
Measure of Similarity
Resume
Job post

Figure 1. High-level overview of the proposed system architecture

As shown in Fig. 1, the proposed system consists of several modules that are organized in the following order. First, the *Concept Identification and Extraction* (CIE) module is employed to create concept lists from both resumes and job posts. Then, the second module of the system refines the lists of concepts through removing those that appear to be of little value and don't have significant meaning in the matching process. By this we mean concepts that express features such as: candidate's name, address, contact info, etc. In addition, concepts that have low tf-idf weights [9] are removed as detailed in section four. Next, the third module of the proposed system takes the refined lists of concepts (from the segments of both the job posts and resumes) as input to construct semantic networks in which concepts are connected by various types of semantic relations (derived from the used semantic resources). During this step, we may find that some concepts are not defined in the exploited semantic resources. These concepts are then submitted to the *Missing Background Knowledge Handler* that relies on another resource (Hiring Solved Dataset [28]) to enrich the constructed semantic networks. In this context, semantically-relevant concepts are extracted and used to expand the constructed semantic networks. Finally, the matching algorithm takes the updated semantic networks as input and produces the measures of semantic closeness between them as output.

IV. Detailed Steps of the Proposed System

Before proceeding to present the details of the methods and techniques used in the proposed system, we introduce – in the context of our work – the terms ''tf-idf Weighting'', "Semantic Resource", "Semantic Network", and "Semantic Network Enrichment".

Definition 1: tf-idf Weighting:
tf_idf weighting scheme [9] assigns a term *t* a weight *w* in a document *d*:

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t \quad (1)$$

It is important to mention that we employ the tf-idf weighting scheme at the corpus level in order to remove the set of concepts S(*cs*) that have no significant meaning among the set of candidate concepts S(*cc*) – obtained using the NLP pre-processing tools introduced in section 4-A. The set S(*cs*) is created based on a threshold value *v* using 2.

$$S(cs,v) := \{\ cs \in S(cc) \mid \text{tf-idf}(cs) \leq v\ \} \quad (2)$$

Definition 2: Semantic Resource:
A semantic resource *SR* is *quintuple, SR*:= <*C, P, I, V, A*> where:

- *C* represents the set of concepts that are defined in *SR*. The hierarchical relationship between concepts of the set *C* is a pair (*C,* ≤*)*, where ≤ is an order relation on *C* x *C*. We call ≤ the sub-concept relation.
- *P* represents the set of properties defined over *C.*
- *I* is the set of individuals also called instances of the concepts in *SR.*
- *V* is the set of values defined over *P.*
- *A is the set of axioms* in *SR.*

For each job post and resume pair, the system takes the extracted lists of concepts as inputs and outputs the following sets of semantic networks:

- The set of semantic networks Sζj = {ζj1, ζj2, ζj3, ζjn} that are automatically derived and constructed from the job post.
- The set of semantic networks SζR = {ζr1, ζr2, ζr3, ζrn} that are automatically derived and constructed from the resume.

As we have pointed out in section 1, to automatically constructed such networks we rely on the exploited semantic resources S = {*SR*1, *SR*2, *SR*3}. We formally define a semantic network as follows:

Definition 3: Semantic Network:
A semantic network ζ:= <CC, RR, AA> where:

- CC is the set of concepts captured by ζ. These are are the resume and job post concepts that are also captured by the semantic resources.
- RR is the set of relations that connect the concepts in CC. Similar to CC, these relations are obtained from the set of semantic resources S = {*SR*1, *SR*2, *SR*3}.
- AA is the set of axioms defined on CC and RR based on each *SR.*

As highlighted in the previous section, the exploitation of more than once semantic resources does not necessarily guarantee full domain coverage. Therefore, we may encounter the problem of unrecognized entities (either concepts or individuals) in the used semantic resources. To tackle this problem, we exploit Hiring Solved Dataset to enrich the semantic networks of both job posts and resumes. Formally, we define the process of semantic networks enrichment as follows:

Definition 4: Semantic Network Enrichment:
The semantic network enrichment process takes a given semantic resource *SR* and a given concept *c* as inputs and produces for c a set S(*c*) ⊑ T(*SR*) as output.
Where:

- S(*c*) is the set of suggested enrichment candidates for c. A candidate t∈T(*SR*) is a single-term or compound-term from *SR*.
- T(*SR*) is the set of entities defined in *SR*.

A. Concept Extraction based on the NLP Techniques

During this step, candidate concept lists of job posts and resumes are identified and extracted based on performing the following NLP steps:

- Document segmentation: the text of a given resume/job post is segmented into paragraphs in order to process each paragraph separately.
- Text tokenization: the text in each paragraph is split into unigram, bigram and trigram tokens.
- Stop words removal: a list of stop words that appear to be of little value in the matching process is defined. Such words are removed to enhance the system's performance and increase its effectiveness.
- Part-of-Speech Tagging: each token is assigned to its part of speech category such as noun, verb, adjective, etc. To accomplish this task, we employ the StanfordCoreNLP POSTagger.
- Named Entity Recognition (NER): during this step, each token is assigned a category based on a set of

pre-defined categories such as person, organization, and location.

The next example clarifies the process of identifying and extracting candidate concepts based on utilizing the abovementioned NLP steps.

Example 1: Extracting Candidate Concepts

- **Part of a job post (P1):**

We are seeking a programmer who is looking to take his experience to the next level. Our programmer is required to have 2+ years of experience in Java programming language (e.g. jsp).

- **Part of an applicant's resume (CV1):**

I have worked as a Software engineer. And I have the following skills: Java, j2ee, jsp, xml.

In this example, we have considered part (one segment) of both the job post and the resume. Accordingly, the text tokenization is performed and stop words are removed. Next, the POS-Tagging and the NER steps are carried out using the StanfordCoreNLP package. In the context of our work nouns (NNP, NNPS, NN) are included in the lists of candidate concepts. The results of applying these steps are shown in Table 1 below.

TABLE I. RESULTS OF APPLYING THE NLP STEPS

Candidate concepts extracted from job post (P1)	Candidate concepts extracted from resume (CV1)
programmer	software engineer
experience	skills
java	java
programming language	j2ee
jsp	jsp
	xml

After extracting candidate concepts, they are refined as detailed in the next section.

B. Refinement of Candidate Concepts

At this step, we use a list of pre-defined terms that represent features such as: candidate's name, contact info, address, birth date, etc. In addition, we utilize the tf-idf weighting scheme to identify concepts that have no significant meaning and may negatively impact the matching process. Accordingly, concepts that either belong to the list of pre-defined terms or have low tf-idf weights are removed from the lists of candidate concepts as illustrated in Table 2.

TABLE II. LISTS OF REFINED CONCEPTS

Job post (P1) concepts list	Resume (CV1) concepts list
Programmer	software engineer
Java	java
programming language	j2ee
Jsp	jsp
	xml

C. Construction of Semantic Networks

In this section, we present the details of constructing semantic networks from the lists of refined candidate concepts.

- **Using WordNet Ontology**

Each concept is submitted to WordNet ontology [27] in order to extract the semantic and taxonomic relations that may exist between it and the other concepts in the list. As a result of this step, semantic networks that represent resumes and job posts are constructed. Figures 2 and 3 depict the output of employing the semantic networks construction module based on WordNet ontology.

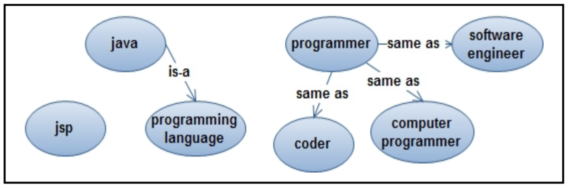

Figure 2. Semantic networks of job post (P1)

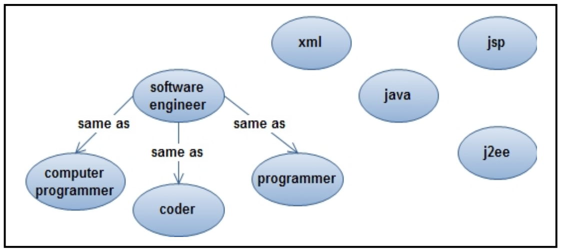

Figure 3. Semantic networks of resume (CV1)

When we look for the term "*java*" (which exists in job post (P1) and resume (CV1)) in WordNet ontology, we can see that it has different senses (i.e. meanings). It is clear that in the context of both (P1) and (CV1) *java* refers particularly to the third sense (*a simple platform-independent object-oriented programming language…*). Therefore, we employ a Word Sense Disambiguation (WSD) technique to specify the correct sense for each term according to its surrounding textual content. Besides, the synonyms of each disambiguated terms are used to expand the constructed semantic networks. The rest of concepts that are missing from WordNet ontology are then submitted to YAGO2 ontology.

- **Using YAGO2 ontology**

Concepts that are not defined in WordNet are then submitted to YAGO2 ontology [29]. Accordingly, semantic relations that are captured in YAGO2 are also exploited to expand the constructed semantic networks. However, we would like to point out that even using a second ontology like YAGO2 may not solve the semantic knowledge incompleteness problem since some concepts such as "jsp" are not defined in it. Therefore, concepts that are not recognized in WordNet and in YAGO2 ontologies are further submitted to the missing background knowledge handler.

D. Missing Backround Knowledge Handler

When the exploited semantic resources fail in recognizing a certain concept, Hiring Solved (HS) dataset [28] is then employed to compensate for such missing background knowledge. HS dataset defines a huge number of terms in the form of skills – either mentioned is job posts or resumes – and the weights of semantic closeness between them. For example, although the term "jsp" was not recognized by the exploited semantic resources, when we submit it to HS dataset we get a set of statistical-based semantically-related terms to "jsp" as shown in Table 3. The weights shown in the table represent measures of semantic relatedness between the submitted term and a set of semantically-close terms to it.

TABLE III. THE RESULT OF SUBMITING "JSP" TO HS DATASET

Term	Relatedness measure
servlets	1.00
j2ee	0.94
Jdbc	0.92
tomcat	0.90
ejb	0.76
struts	0.75
hibernate	0.62
xml	0.60
java	0.56

Following to this step and based on the results of applying the missing background knowledge module, the semantic networks are updated and expanded as depicted in Figures 4 and 5.

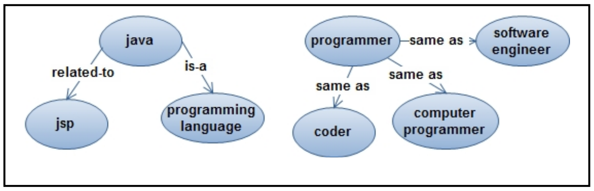

Figure 4. Updated semantic networks of job post (P1)

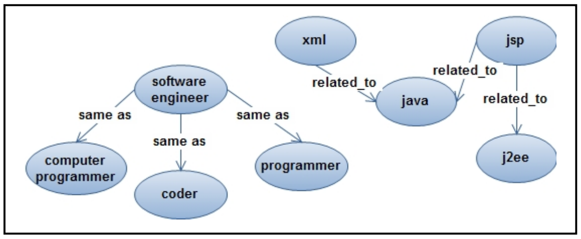

Figure 5. Updated semantic networks of resume (CV1)

As shown in Figures 4 and 5, concepts in the semantic networks of (P1) and (CV1) are linked with the newly obtained concepts from HS dataset. For instance, we can see that the degree of semantic closeness between the terms "j2ee" and "jsp" is 0.94. We then replace this semantic relatedness value by the "related-to" relation and use it to connect both concepts.

E. Further Enrichment of the Semantic Networks

Semantic networks extracted from job posts represent the reference to which the semantic networks of the resumes are matched. In this context and since some of the required skills may not be explicitly defined by the employer, we further enrich the semantic networks of the job posts by automatically adding new skills obtained from HS dataset. To carry out this step, we submit the job titles to HS dataset to obtain a set of related skills to each title. For instance, when submitting the job title ("java programmer") of job post (P1) to HS dataset, it returns the list of skills shown in Fig. 6. As highlighted in the previous section, we replaced the measures of semantic relatedness with the "related-to" relation and only considered the top 5 related skills returned by HS dataset.

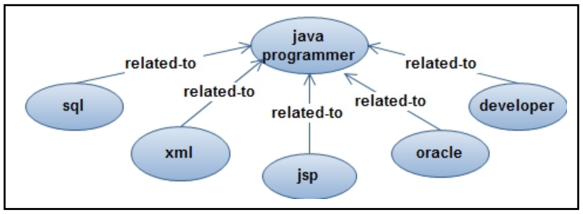

Figure 6. Top 5 related skills returned by HS dataset to the title "Java Programmer"

To enrich the semantic networks of job post (P1) with the elements of S(*cr*), we follow the following procedure:

- If an element *cr* ∊ S(*cr*) already exists in the semantic networks of (P1), then we retain *cr* in its position in the networks. For example, since the element "jsp" is already defined in the semantic networks of (P1), we keep this element in its position in the network.
- If an element *cr* ∊ S(*cr*) does not exist in the semantic networks of (P1), then we update the networks by adding the job title as a new node and then attaching it to all other elements of S(*cr*) that do not exist in the semantic networks of (P1).

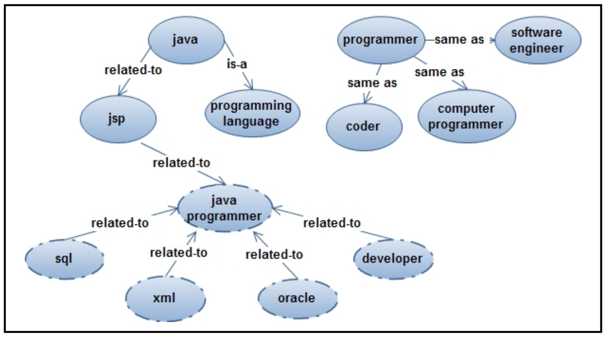

Figure 7. Enrichment of the semantic netowrks of job post (P1)

F. *Matching the Semantic Networks*

During this step, the semantic networks of the resumes and job posts are matched based on the Jaro-Winkler edit distance function. This function returns a measure of similarity between the strings of the nodes in the semantic networks. In this context, the higher the similarity between the nodes, the more a job post and resume pair are considered relevant to each other.

Algorithm 1. Edit-distance algorithm for computing the similarity between SN_R (Semantic Network of Resume R) **and SN_J** (Semantic Network of Job Post J)

```
Input: $SN_R$ and $SN_J$
Output: Measure of string similarity based on the set of correspondences $S$
1: int similarity;
2: $result \leftarrow \langle \rangle$;
3: for i=0; i < $SN_R$ .Length; i++
4:   for j=0; j < $SN_J$ .Length; j++
5:     $result$=Jaro-Winkler($SN_R$ [i], $SN_J$ [j])
6:     if(result< v) then
7:       add($SN_R$[i], $SN_J$[j]) to $S$
8:       similarity++;
9:     end if
10:  end for
11: end for
12: return $result$
```

We employ the above algorithm to compute the string dissimilarity between two strings (line 5). If the result is less than a threshold value *v*, then the two strings are considered as equivalent and added to the set of correspondences *S*. As such, each resume is ranked based on the returned similarity values between its semantic networks and the semantic networks of the job post.

V. Experimental Results

This section describes the experiments carried out to evaluate the techniques of the proposed system. A prototype of the proposed system is implemented and experiments are conducted using a PC with dual-core CPU (2.1GHz) and (4 GB) RAM. The used operating system is Windows 7.

To measure the effectiveness of the proposed system, we evaluated its precision in assigning relevance scores between job posts and applicant resumes. To accomplish this task, we collected a data set of 500 resumes downloaded from http://www.amrood.com/resumelisting/listallresume.htm and other local job portals, and used ten different job posts obtained from http://jobs.monster.com. The collected resumes and job posts are unstructured documents in different document formats such as (.pdf) and (.doc). In order to carry out the experiments, we analyzed the corpus of resumes and job posts through employing the NLP tools described in section four. Then, we utilized statistical-based concept-relatedness measures to refine the lists of candidate concepts. Next, we used the semantic resources to construct the semantic networks of job posts and resumes. Additionally, the constructed networks were further enriched based on HS dataset. And finally, the resulting networks were automatically matched and different relevance scores were produced by the system.

A. *Experiements Using Expert Judgments*

In order to provide a ground for evaluating the quality of the results produced by the system, we manually calculated all relevance scores between each job post and its relevant resumes. Then, we compared the manually calculated scores to those produced by the system. In this context, we used the Precision (P) indicator in order to measure the quality of our results. This measure is defined as follows:

Precision (P): is the Percentage Difference between the manually assigned relevance scores (between each job post and its relevant resumes) and those automatically generated by the system.

$$P = \frac{|V_{manual} - V_{automatic}|}{(\frac{V_{manual} + V_{automatic}}{2})} * 100\% \qquad (3)$$

where:

- V_{manual} : is the manually assigned relevance score between each resume and job post.
- $V_{automatic}$: is the automatically calculated relevance score between each resume and job post.

TABLE IV. Precision Results of the Automatically Generated Relevance Scores by the System

Job post	Resumes	Manual score	Automatic score	P (%)
Programmer	IT-QA	0.20	0.28	0.77
	IT-Programming	0.40	0.41	0.97
	IT-Tele-Software	0.70	0.68	0.97
	Graphics and web design	0.08	0.09	0.88
	Network admin	0.10	0.16	0.63
	Software engineer	0.05	0.05	1.00
.Net developer	IT-QA	0.90	0.70	0.75
	IT-Programming	0.40	0.44	0.91
	IT-Tele-Software	0.50	0.80	0.63
	Graphics and web design	0.10	0.12	0.83
	Network admin	0.21	0.41	0.46
	Software engineer	0.10	0.17	0.49
Database developer	IT-QA	0.30	0.42	0.77
	IT-Programming	0.44	0.50	0.88
	IT-Tele-Software	0.70	0.62	0.88
	Graphics and web design	0.20	0.22	0.91
	Network admin	0.40	0.44	0.90
	Software engineer	0.07	0.07	1.00

As shown in Table 4, for each job post, we compared between the manually assigned relevance score for each resume and its corresponding relevance score that is automatically produced by the system. We considered six resumes per job post. Each job post requires a different set of qualifications. The first job post requires skills in java, jsp, jsf, html, and javascript, and five years of experience. The second job post divides the required qualifications into two categories: i) Obligatory: having 6-8 years of experience in developing web applications using .Net technologies (asp.net, c#, mvc vb.net, etc) and ii) Optional: having experience in jquery, vb Script, and ajax. The third job post focuses on Microsoft sql related skills. As we can see in Table 4, the manual scores that were assigned for each resume by our expert are very close to the automatically calculated scores by the system. This is due to the fact that we employ multiple semantic resources that represent the semantic aspects of resumes and job posts. Additionally, we exploit statistical concept-relatedness measures to compensate for missing background knowledge and to enrich the list concepts that are extracted from the job posts with relevant concepts that were not recognized by the semantic resources.

However, we can find that for some particular results the percentage difference was large. For example, when matching the second job post ".Net developer" and IT_QA resume, the difference is (0.25 i.e. 100% - 75%). This is because the job post ".Net developer" has optional requirements in its job description such as (having experience in jquery, vb Script and ajax). This optional requirement is not distinguished from other obligatory requirements by our system and thus the manual score for the resume is larger than the automatic score. In order to solve this problem, we plan to assign different weights for optional and obligatory requirements, and then use these weights in computing the relevance scores between job posts and resumes.

B. *Evaluating the System's Effectiveness when Utilizing the Statistical Concept-Relatedness Measures*

In this section, we compare between the produced results by the proposed system when we utilize the statistical concept-relatedness measures against when only using the semantic resources. We used the **Precision/Recall** (P/R) indicators in order to evaluate the quality of the produced results where:

$$P = \frac{|\{relevant\ resumes\} \cap \{retrieved\ resumes\}|}{|\{retrieved\ resumes\}|} \qquad (4)$$

$$R = \frac{|\{relevant\ resumes\} \cap \{retrieved\ resumes\}|}{|\{relevant\ resumes\}|} \qquad (5)$$

TABLE V. PRECISION/RECALL RESULTS USING/NOT USING THE STATISTICAL TECHNIQUES

	Job post	P/R Results without using the statistical techniques		P/R Results using the statistical techniques	
		P	R	P	R
1	Programmer	0.41	1.0	1.0	0.83
2	Java software engineer	0.71	0.69	0.88	0.97
3	Database developer	0.35	0.83	0.9	0.75
4	Senior QA enginer	0.45	0.91	0.84	0.91
5	Software quality engineer	0.48	1.0	0.82	1.0
6	Senior database administrator	0.23	0.8	1.0	0.8

As shown in Table 5, we were able to achieve promising precision results for most of the job posts. Additionally, it was obvious that a significant improvement on the produced results was achieved when utilizing the statistical-based concept-relatedness techniques. This is because when using these techniques we were able to refine the lists of candidate concepts on the one hand, and further enrich them with more related concepts on the other.

C. *Comparison with State-of-the-Art Systems*

In this section, we compare the results produced by our systems with EXPERT system [14] which is one of the state-of-the-art semantics-based automatic recruitment systems. Both systems were tested against the dataset obtained from http://www.amrood.com/resumelisting/listallresume.htm.

TABLE VI. PRECISION/ RECALL RESULTS

System Indicator	Our system	EXPERT
P	0.91	0.89
R	0.88	0.93
F-measure	0.89	0.87

As shown in Table 6, our proposed system was able to achieve better results than EXPERT system. The reason behind this is that – unlike EXPERT system – we are exploiting multiple semantic resources to derive the semantic aspects of resumes and job posts. In addition, we utilize Hiring Solved dataset to compensate for missing background knowledge and to enrich job posts with skills that are not explicitly mentioned by the employer. It is important to mention that we will incorporate other features in the matching algorithm to improve the effectiveness of the proposed system. Accordingly, we will integrate a features extraction module to extract features such as educational background and years of experience from applicants' resumes. We believe that incorporating these features will lead to improving the results produced by the system.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed an automatic online recruitment system based on coupling multiple semantic resources and statistical concept-relatedness measures. The proposed system first employs NLP techniques to identify and extract candidate concepts from job posts and resumes. Then, statistical-based concept-relatedness measures are utilized to refine the lists of extracted concepts from the job posts and resumes. Next, the system employs multiple semantic resources to derive the semantic aspects of resumes and job posts. Since these semantic resources are limited in terms of their domain coverage, we use HS dataset to address this limitation and to enrich the job posts with further semantically-related concepts. As indicated in section 5, the initial experiments using different resumes and job posts showed promising precision results. In the future work, we plan to test the proposed system using additional resumes and job posts. Besides, we plan to integrate a features extraction module to the proposed system in order to extract other important features such as educational background and years of experience from resumes.

REFERENCES

[1] S. Strohmeier, and F. Piazza, "Domain driven data mining in human resource management," A review of current research. Expert Syst. Appl., vol. 40(7), 2013. pp. 2410-2420.

[2] M. Kessler, et al., "A hybrid approach to managing job offers and candidates," Inf. Process. Manage., vol. 48(6), 2012, pp. 1124-1135.

[3] C. Bizer, et al., "The impact of semantic web technologies on job recruitment processes," in Wirtschaftsinformatik. Physica-Verlag HD, 2005, pp. 1367-1381.

[4] M. Mochol, H. Wache, and L. Nixon, "Improving the Accuracy of Job Search with Semantic Techniques," in Business Information Systems, W. Abramowicz, Editor, Springer Berlin Heidelberg, 2007, pp. 301-313.

[5] F. García-Sánchez, et al., "An ontology-based intelligent system for recruitment," Expert Systems with Applications, vol. 31(2), 2006, pp. 248-263.

[6] S. Colucci, et al., "A formal approach to ontology-based semantic match of skills descriptions," J. UCS, vol. 9(12), 2003, pp. 1437-1454.

[7] L. Sivabalan, Y. Rashad, and I. Nor Haslinda, "How to Transform the Traditional Way of Recruitment into Online System," International Business Research, vol. 7 Issue 3, 2014, pp. 178-185.

[8] S. Pande, "E ‐ recruitment creates order out of chaos at SAT Telecom," Human Resource Management International Digest, vol. 19(3), 2011, pp. 21-23.

[9] N.J. Belkin and W.B. Croft, "Information filtering and information retrieval: two sides of the same coin?," Commun. ACM, vol. 35(12), 1992, pp. 29-38.

[10] R. Kessler, et al., "Job Offer Management: How Improve the Ranking of Candidates," in Foundations of Intelligent Systems, J. Rauch, et al., Editors., Springer Berlin Heidelberg, 2009, pp. 431-441.

[11] E. Faliagka, Ramantas, K., Tsakalidis, A., Viennas, M., Kafeza, E., & Tzimas G. (2011), "An Integrated E-Recruitment System for CV ranking based on AHP," in the Proceedings of the 7th Web Information Systems and Technologies (WEBIST 2011), 6-9 May, Netherlands, SciTePress, ISBN 978-989-8425-51-5, 2011, pp. 147-150.

[12] F. Trichet, et al., "Human resource management and semantic Web technologies," in Information and Communication Technologies: From Theory to Applications, 2004. pp. 641-642.

[13] M. Mochol, E. Paslaru, and B. Simperl, Practical guidelines for building semantic erecruitment applications. International Conference on Knowledge Management, Special Track: Advanced Semantic Technologies (AST'06), 2006, pp. 1-8.

[14] V.S. Kumaran, and A. Sankar, "Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping EXPERT," Int. J. Metadata Semant. Ontologies, vol. 8(1):, 2013, pp. 56-64.

[15] S., Chung-Kwan, et al., "A hybrid approach of neural network and memory-based learning to data mining," Neural Networks, IEEE Transactions on, vol. 11(3), 2000, pp. 637-646.

[16] M. Kessler, et al., "E-Gen: automatic job offer processing system for human resources," in Proceedings of the artificial intelligence 6th Mexican international conference on Advances in artificial intelligence, Springer-Verlag: Aguascalientes, Mexico, 2007, pp. 985-995.

[17] E. Faliagka, et al. "Application of machine learning algorithms to an online recruitment system," in ICIW 2012, The Seventh International Conference on Internet and Web Applications and Services. 2012. pp. 215-220.

[18] E. Faliagka, A. Tsakalidis, and G. Tzimas, "An integrated e-recruitment system for automated personality mining and applicant ranking," Internet research, vol. 22(5), 2012, pp. 551-568.

[19] Hong, W., et al., "A Job Recommender System Based on User Clustering," Journal of Computers, vol. 8(8), 2013, pp. 1960-1967.

[20] M. Maree, and M. Belkhatir, "Addressing semantic heterogeneity through multiple knowledge base assisted merging of domain-specific ontologies," Knowl.-Based Syst, vol. 73, 2015, pp. 199-211.

[21] I. Lee, "An architecture for a next-generation holistic e-recruiting system," Commun. ACM, vol. 50(7), 2007, pp. 81-85.

[22] V. Lavrenko, and W.B. Croft, "Relevance based language models," in Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, ACM: New Orleans, Louisiana, USA, 2001, pp. 120-127.

[23] X. Yi, J. Allan, and W.B. Croft, "Matching resumes and jobs based on relevance models," in Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, ACM: Amsterdam, The Netherlands, 2007, pp. 809-810.

[24] M. Maree, and M. Belkhatir, "A Coupled Statistical/Semantic Framework for Merging Heterogeneous Domain-Specific Ontologies," in 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), 2010, pp. 159-166.

[25] V. Senthil Kumaran, and A. Sankar, "Expert locator using concept linking," International Journal of Computational Systems Engineering, vol. 1(1), 2012, pp. 42-49.

[26] K. Ramar, and N. Sivaram, "Applicability of clustering and classification algorithms for recruitment data mining," Int J Comput Appl, vol. 4(5), 2010, pp. 23-28.

[27] G.A. Miller, "WordNet: a lexical database for English," Commun. ACM, vol. 38(11), 1995, p p. 39-41.

[28] Hiring Solved Dataset. Available from: https://hiringsolved.com/explorer.

[29] J. Hoffart, et al., YAGO2: exploring and querying world knowledge in time, space, context, and many languages, in Proceedings of the 20th international conference companion on World wide web, ACM: Hyderabad, India, 2011, pp. 229-232.

[30] W.E. Winkler, "The State of Record Linkage and Current Research Problems," Statistics of Income Division, Internal Revenue Service Publication, 1999.