7-17-2019

A New Approach to Information Extraction in User-Centric E-Recruitment Systems

Malik Nabeel Ahmed Awan

Sharifullah Khan

Khalid Latif

Asad Masood Khattak

applied sciences

MDPI

Article

A New Approach to Information Extraction in User-Centric E-Recruitment Systems

Malik Nabeel Ahmed Awan [1,*], Sharifullah Khan [1], Khalid Latif [2] and Asad Masood Khattak [3]

[1] SEECS, National University of Sciences and Technology, Islamabad 44000, Pakistan
[2] Biome Analytics, Islamabad 44000, Pakistan
[3] College of Technological Innovation, Zayed University, Abu Dhabi 144534, UAE
* Correspondence: nabeel.ahmed@seecs.edu.pk

Received: 26 June 2019; Accepted: 15 July 2019; Published: 17 July 2019

Abstract: In modern society, people are heavily reliant on information available online through various channels, such as websites, social media, and web portals. Examples include searching for product prices, news, weather, and jobs. This paper focuses on an area of information extraction in e-recruitment, or job searching, which is increasingly used by a large population of users in across the world. Given the enormous volume of information related to job descriptions and users' profiles, it is complicated to appropriately match a user's profile with a job description, and vice versa. Existing information extraction techniques are unable to extract contextual entities. Thus, they fall short of extracting domain-specific information entities and consequently affect the matching of the user profile with the job description. The work presented in this paper aims to extract entities from job descriptions using a domain-specific dictionary. The extracted information entities are enriched with knowledge using Linked Open Data. Furthermore, job context information is expanded using a job description domain ontology based on the contextual and knowledge information. The proposed approach appropriately matches users' profiles/queries and job descriptions. The proposed approach is tested using various experiments on data from real life jobs' portals. The results show that the proposed approach enriches extracted data from job descriptions, and can help users to find more relevant jobs.

Keywords: semantic web; information retrieval; information extraction; e-recruitment

1. Introduction

With recent advancements in technology, human reliance on the internet has increased greatly. Information is now mostly available and shared via the internet using sources, such as websites, social media and web portals. This advancement in internet technology has also had an impact on recruiting potential employees for an organization.Various e-recruitment systems have flourished, such as Indeed (https://www.indeed.com), Monster (https://www.monster.com/), Person force (https://www.personforce.com/), and Angel.co (https://angel.co/). E-recruitment facilitates both employers and users in terms of finding relevant jobs efficiently. Existing e-recruitment systems use keywords or faceted searches (https://www.indeed.com; https://www.monster.com/; https://www.personforce.com/; https://angel.co/) to provide better search results to both organization and users. The recruitment process starts with advertising a job description; Figure 1 shows a sample job description with its segments.

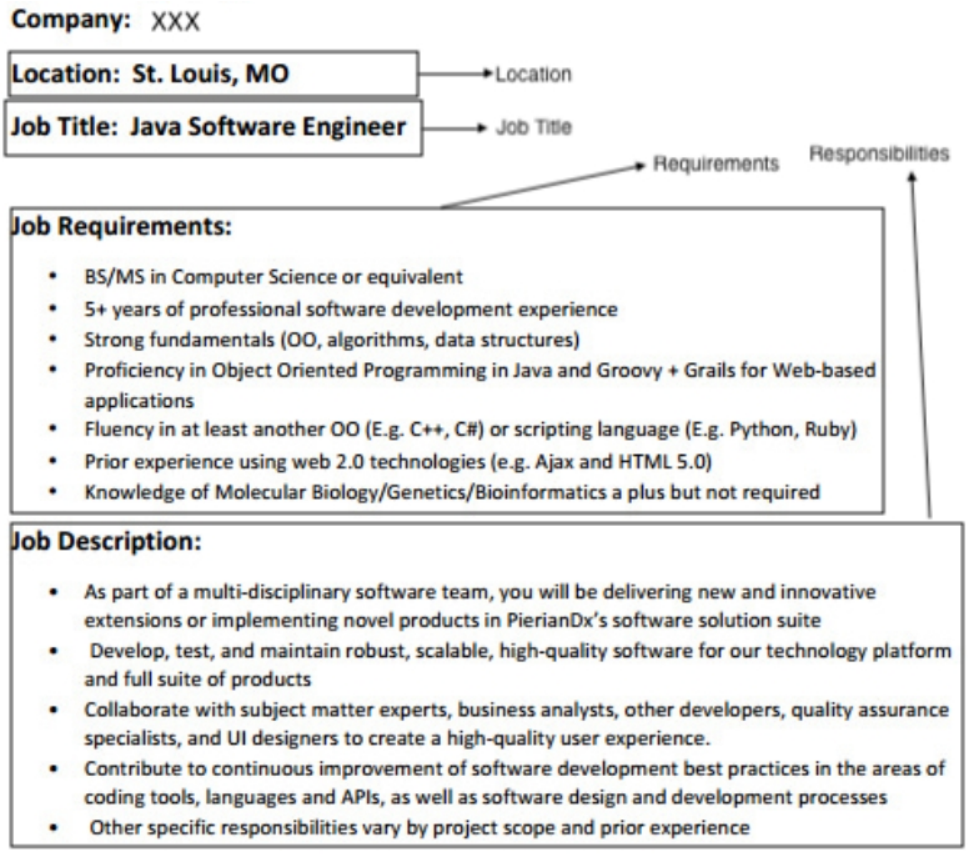

Figure 1. A sample Job description with marked segments.

The job description in Figure 1 outlines the location, job title, requirements and responsibilities. Some features in a job description, such as location, job title, skills and expertise level, are described as entities. These entities are contextually associated with each other to yield contextual entities, such as the job requirements. The employer's primary emphasis in candidate filtering is on the job requirements because the requirements define the baseline for the selection of a potential candidate. The job description/candidate profile contents thus hold crucial importance. However, the information provided in the job description/user profile provides challenges for extraction, such as the fact that content is unstructured; there is no standard format for defining content, and there are text nomenclature differences for defining the same content. Recruitment processes prioritize matching/relevance between job description and candidate queries to filter out irrelevant candidates, or get the most well-fitting job for the candidate. Manually performing this matching process is time-consuming and challenging [1], and the process is carried out automatically in e-recruitment systems. However, this process is not merely the matching of text because there can be semantic heterogeneity in the texts. For example, in Figure 2, the text of a job description and user query have been illustrated. There is no match in them lexically; however, they are semantically matching because 'Android Development' is a type of 'mobile application'. The matching process is complex and needs to understand the context (i.e., domain-specific information) of the text to resolve semantic heterogeneity in matchmaking.

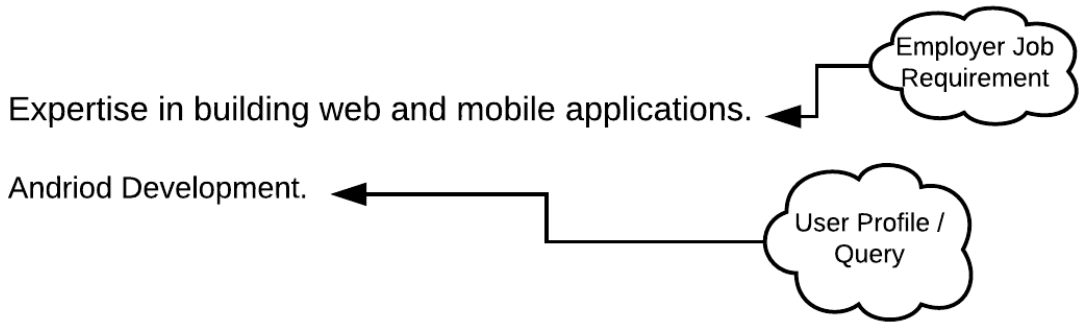

Figure 2. A sample semantic difference between a job description and a user profile/query.

Existing e-recruitment systems [2,3] do not extract domain-specific information, such as 'mobile application' in the above example from the job requirement text to match with the candidate query of 'Android Development'. These domain-specific entities are contextually associated with each other. Existing systems extract entities from text independently of each other, without considering the context associated with the entities. For example, the 3rd job requirement in Figure 1 'Strong fundamentals (OO, algorithm, data structure)' has the entity 'trong' as the expertise level and 'OO, algorithm, data structure' as skills, but actually 'strong' expertise is required in 'OO, algorithm, data structure'. These entities combine to generate a contextual entity, i.e., a job requirement. Another drawback of existing systems is that of the limited availability of information [4,5] contained in the knowledge base for enrichment either through in-house data or with an external source that is static with respect to data growth. On the other hand, Linked Open Data (LOD) [6] does not suffer from data staleness and data can expand over time. Multiple sources are actively contributing to it, such as Wikipedia (https://www.wikipedia.org/), Getty (http://www.getty.edu/research/tools/vocabularies/lod/), and GeoNames (https://lod-cloud.net/dataset/geonames-semantic-web). Existing approaches [7,8] do not properly implement LOD principles.

The proposed e-recruitment system in this paper, called SAJ, addresses all the issues mentioned above. First of all, it divides the job description into segments, such as location, job title, and requirements. Then, it extracts domain-specific entities, such as the job title, location, expertise level, and career level, from the job description, and builds relationships among entities to extract contextual entities, such as the job requirement and job responsibilities. The system enriches entities using Linked Open Data to make them able to resolve semantic heterogeneities, such as that shown in Figure 2. The context-aware information is stored in the knowledge base built using Linked Open Data principles. The proposed system improves the user experience for both employer and candidates. It has been evaluated against some well-known information extraction systems and tools. SAJ performs well against these other systems and tools.

This paper is organized as follows: Section 2 outlines the related work in information extraction and e-recruitment, Section 3 presents the proposed e-recruitment system (SAJ), Section 4 explains the extraction and enrichment process with examples, Section 5 discusses the evaluation and finally Section 6 concludes the paper.

2. Related Work

The focus of SAJ is on automatic extraction of contextual entities, such as job requirements and job responsibilities, from an unstructured job description. The extracted entities are then enriched and connected using a job description ontology to build context. The extraction of entities from a job description requires a comprehensive understanding of its structure and semantics. It involves in-depth knowledge of multiple domains that range from information extraction, knowledge base population, Linked Open Data and most importantly e-recruitment systems. Each of these domains has significance in SAJ. E-recruitment is an overall domain that defines boundaries [3,9] of our work

and knowledge of e-recruitment is critical in the overall definition of SAJ. These relevant topics will be explored in the following section to put into context their state-of-the-art approaches.

2.1. Information Extraction

Information extraction is a technique for identification of entities from unstructured text [10]. A job description contains entities and compound words, such as job titles, job requirements, job responsibilities, career level and others entities that need to be extracted. Extraction techniques, to cater for unstructured document formats, mainly focus on rules and pattern [11], machine learning [12] and ontology [13] based approaches.

Rule and pattern-based techniques identify hidden features in the text by utilizing predefined rules or known patterns [11], e.g., extraction of a person's phone number may require the occurrence of phrases, such as "at", "can be (reached | called) at" and "s number is". Absence of these phrases will result in the non-extraction of the person's phone number. Rule-based techniques have been applied in multiple domains, such as aspect extraction of product reviews by exploiting common-sense knowledge and sentence dependency trees [14], relation extraction using background knowledge [15], extraction of patient's clinical data from medical texts [16] and numerous others. These methods may process unstructured text multiple times to get meaningful information. Besides this, rule based techniques have been applied in the extraction of compound entities from bio-medial domains [17] using BioInfer and GENIA corpuses. The drawback of compound word extraction using the technique mentioned by Cartic Ramakrishnan et al. [17] is that any concept that is missing in the BioInfer and GENIA corpuses will not be identified as a compound word. These methods cannot identify any new rule or pattern that is not already defined. Machine learning based techniques can deal with such problems.

Machine learning based techniques help in the extraction of existing and new information from unstructured texts. Machine learning techniques use Hidden Markov Models (HMM) [18] and Conditional Random Fields (CRF) [19] to extract information from unstructured text. Machine learning based techniques require large data-sets for training and evaluation purposes. The aforementioned techniques fail to link information together with context. The lack of context may result in information loss.

Ontology based techniques cover this gap in information extarction, and mainly use domain-specific knowledge for extracting meaningful information from unstructured text [20]. Some well known existing systems that use domain ontology are KIM [21] and TextPresso [22]. These systems only use information present in domain ontology to facilitate entity extraction. TextPresso mainly focuses on entity extraction in the bio-medical domain. It uses Gene Ontology (GO) during extraction that comprises approximately 80% of the lexicon. Any new information extracted will result in information loss. This limitation was addressed by the technique proposed by Vincent et al. [13]. According to their technique, the newly extracted knowledge is merged with existing domain knowledge resulting in enhanced domain knowledge. Further down the road, information extraction has also been supported with fuzzy ontology [23] and used in extracting travelers' reviews about hotels, building and designing business intelligence systems for gathering company intelligence and country/region information [24], building and designing systems to extract information from clinical documents such as admission reports, radiology findings and discharge letters [25], and a framework for retrieval of images from web data [26].

The technique proposed by Vincent et al. [13] has the limitation of only using the lexical English database WordNet (https://wordnet.princeton.edu/) as an external source for enhancing information. The enhancement was limited to WordNet which suffers a staleness issue from data. This issue of data staleness has been addressed in other studies by [27–29], which have updated domain ontology independently of WordNet, thus increasing the extraction precision. They have used the pattern-based approach and ontology for extracting new concepts that are not modeled in domain ontology, thus enriching the ontology.

2.2. Knowledge Base Construction

A knowledge base stores the extracted entities along with their relations from a job description(s). Knowledge Base Construction (KBC) explores techniques for populating knowledge bases with rich information either from unstructured, semi-structured or structured documents. KBC is an iterative process [7] that creates linkages between entities, such as skills: Object Oriented and expertise level: strong.

The system proposed by Gregoet et al. [8] automatically populates a knowledge base from both structured and unstructured text using an ontology. This technique can be applied to any field provided its ontology. Another system KELVIN [30] extracts entities and relations from large text collections. The core features of KELVIN are (1) cross-document entity co-references, (2) inter-document co-reference chains, (3) a slot value consolidator for entities, (4) the application of inference rules to expand the number of asserted facts and (5) a set of analysis and browsing tools supporting development.

Deepdive [7,31] is also an effort for populating knowledge bases from dark data [32]. The deep-dive approach utilizes database and machine learning techniques. Deepdive extracts structured data from unstructured dark data that includes the mass of text, tables, and images that are collected and stored but which cannot be queried using traditional relational database tools. From a machine learning perspective, Deepdive's language inherits Markov Logic Networks [33].

Another focus for KBC is to extract organizational email entities from the Enron data-set (https://www.cs.cmu.edu/~./enron/) [34]. This data-set contains about 0.5 M emails from about 150 senior management users, employed by Enron. This technique's focus is to populate an organizational KB from an email collection, and extend an existing entity linking evaluation collection. It links the organizational email mentions detected from sampled Enron email messages to both Wikipedia and new domain-specific KBs.

The techniques mentioned above, such as those mentioned in Refs. [7,8,34], have not been developed using Linked Open Data principles and also may not be able to facilitate data storage using Linked Open Data principles (https://ontotext.com/knowledgehub/fundamentals/linked-data-linked-open-data/).

2.3. Data Enrichment and Linked Open Data

Data enrichment is the process of enhancing, refining or improving existing data. The enrichment process increases the data's value [35]. Work has been carried out in various domains, such as cultural heritage [5], scientific publications [36], question answering [37], web pages [38] and others areas to enhance, refine and improve existing data by adding more knowledge from external sources. The process is gaining increasing popularity [4]. Significant work has been carried out on experimental data [4]. A large quantity of scientific and experimental data are produced in the activities of large-scale international campaigns. These data are a fundamental pillar of the scientific and technological advancement of information retrieval [4]. Our proposed system semantically annotates and interlinks the data. The data are then exposed as Linked Open Data (LOD). The interconnections of data in LOD provides a means of data enrichment, i.e., the depth and breadth of the LOD graph increase. Other work has been carried out only solely to enrich data pertaining to cultural heritage [5]. The cultural heritage institute is now progressing towards sharing knowledge via LOD. Sharing data using LOD increases the data's value. The current system connects the Biblioteca Virtual Miguel de Cervantes records (200,000 entities) to other data sources on the web. At the moment, the focus is on enriching location and date information with additional knowledge; the current system uses the GeoNames API to link the data.

Linked Open Data (LOD) [39] is an emerging research area with an immense amount of work being carried out in this domain. This work mainly focuses on the construction of LOD data sources which can be utilized for data sharing, data enrichment and bridging the gap between data interoperability and heterogeneity. Yamaguchi et al. [40] have used LOD for computing class–class relationships and

Koho et al. [41] have used LOD to build the datasets of war history casualties. This was represented as a linked model so that citizens investigate what happened to relatives and the model can be used for research in the digital humanities (DH). Another effort has been made by Maulik et al. [42] for Linked Graph Analytics in Pharmacology. The semantic web community has linked and published several datasets in the life sciences domain as Life Science Linked Open Data (LSLOD) using established W3C rules.

The techniques of LOD and enrichment, as mentioned earlier, are mainly focused on the construction of Linked Open Data sources. The utilization of Linked Open Data will enable a more open space for enriching information extracted from data sources instead of the limited external knowledge provided to them.

2.4. E-Recruitment Systems

Multiple systems have been developed to support the process of e-recruitment. JobOlize [9] is one such system that extracts structured information from unstructured job documents, such as job title, contact details and others. It utilizes a hybrid approach to combine existing Natural Language Processing (NLP) techniques with the new form of context-driven extraction techniques for extracting the layout, structure and content information of a job description. Owoseni et al. [2] built a 3-tier technique using a semantic model. This technique performs analysis using document retrieval and natural language processing techniques for a human-like assessment. Besides these systems, various existing techniques, such as collaborative filtering, content-based filtering, knowledge based, and hybrid approaches [43], are adapted for the candidate and job recommendations. The SEEMP [3] project coordinates between public and private employment services around EU member states. It utilizes a mixed approach to services and semantics. The approach adopted by SEEMP combines software engineering and semantic web methodologies and tools for IT architectures, allowing meaningful service-based communication between employment services. The primary objective of the SEEMP project among European Union (EU) member states was to enhance interoperability for better employment services. To achieve interoperability, SEEMP adapts a Web Services Modeling Ontology (WSMO) [44] for semantic web services, a Web Services Modeling Language (WSML) [45] to encode the descriptions and METHONTOLOGY [46] as a methodology for developing and maintaining semantic descriptions. Other work has been undertaken by Malherbe et al. [47] to categorize job offers. It adopts a bottom-up approach to job categorization; the method detects characterizing terms in a corpus of job offers leading to more effective classification and the approach was evaluated on real-world data by multi-posting, on a sizeable French e-recruitment platform. SCREENER [48] is another e-recruitment system that facilitates the recruitment process by extracting information only from the resumes. It identifies text segments that have a probability of possessing specific information including skills, education, experience, and other related information. A predefined corpus assists in data extraction. The extracted information is then indexed using Lucene (http://lucene.apache.org/) for searching and ranking all applicants for a given job opening. The authors claim that this automated system makes the screening task more straightforward and more efficient.

2.5. Critical Analysis

After reviewing existing systems, it is clear that the following points need attention for employee recruitment systems to be improved.

1. Information loss in the extraction of domain specific e-recruitment entities for job descriptions due to unavailability of their context, or inter and intra document linkages.
2. Information loss due to the absence of a comprehensive schema level domain ontology in e-recruitment for building relationships (hierarchical and associative) among extracted entities for job descriptions.
3. Usage of static sources for data enrichment (expansion) resulting in data staleness.

3. The SAJ System

The proposed human e-recruitment system, called SAJ, exploits Linked Open Data (LOD), job description domain ontology, and domain-specific dictionaries for extraction of entities. The extracted entities are enriched and connected to minimize the loss of information in the extraction process. SAJ combines various operations to achieve extraction and enrichment from the job description in the e-recruitment system. As shown in Figure 3, unstructured job description text extracted from any document format, such as MS Word or a PDF, is input. The raw text is extracted using Apache Tika (https://tika.apache.org/). Then, the text is segmented into predefined categories using a self-generated dictionary. Natural Language Processing (NLP) and the dictionary help in the identification of entities. The entities are forwarded to two parallel processes, context building and entity enrichment. The output of both these processes is integrated and stored in the knowledge base. Extracting entities from the unstructured text is non-trivial and challenging work [49]. SAJ not only extracts entities from job descriptions, but it also enriches them contrary to existing e-recruitment systems [9,44]. The entities extracted by SAJ and their connections can facilitate in searching and retrieval, scoring and ranking of candidates against job description. The following sub-sections present the SAJ components in detail.

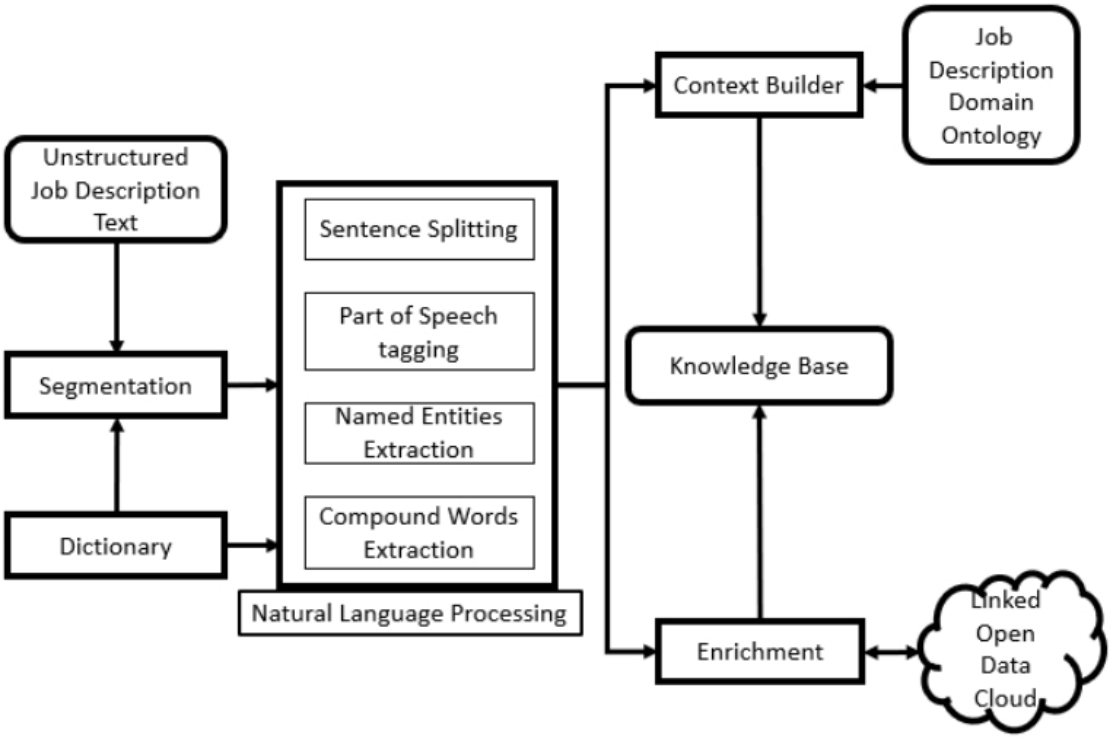

Figure 3. The proposed e-recruitment system SAJ.

3.1. Dictionary

The purpose of the dictionary is to assist in text segmentation and entity extraction. The dictionary is a combination of rules and lists designed for identifying segments and entities in a job description. The rules were composed in Java Annotation Pattern Engine (JAPE) [50]. Table 1 shows sample rules for segmentation and entity extraction along with description.

Table 1. Sample rules designed for segmentation and extraction with description.

Segmentation Rules	
Rule	**Description**
text.sentence.index == 1	Job title as the first line of text
text.sentence.token < 4	Heading line has no other text
Extraction Rule	
Rule	**Description**
Rule:expDurationForSkill (Token.kind==number Token.string==“+”SpaceToken (Token.string==“years” |Token.string==“yrs”)):exp –> :exp.ExpDuration = rule = “expDurationForSkill”	The rules detects experience for a skill, e.g., 2+ years of experience is required in Java

Typically, each rule has two parts, the Left (L.H.S) and the Right (R.H.S). The L.H.S contains the inputs (the identified annotation pattern). They may contain regular expression operators (e.g., *, ?, +). The R.H.S is the rule outcome that is one or multiple annotations to be created based on the L.H.S. Not all rules in the dictionary are applied equally; instead, rules are applied in order of priority. The extraction rule shown in Table 1 extracts a cardinal (CD) number, e.g., 2 describes candidate's level of experience for a particular skill. This process reduces the chances of false positives and also provides a way to verify any error that arises during requirements' boundary identification.

3.2. Segmentation

Segmentation is a process that categorizes text in a job description. The primary objective of segmentation is to ensure that the extracted entities are correct, and belong to the correct text segment. The text segments in the job description are marked with a starting and ending index location.

The text is placed in predefined categories, such as job title, requirements, responsibilities, and career level, among others. At the moment, a dictionary-based approach [48] is adopted for identification of text categorization. The dictionary contains an extended list of possible rules and headings values that can occur in a job description. The rules mentioned in the dictionary ensure that a split is correct along with its category. Figure 1 shows the categories of a job description that are identified by the segmentation process.

3.3. Entity Extraction

The segmented job description is forwarded for entity extraction. With the help of the NLP technique, this component first extracts all sentences from the text and then marks each token/word with its respective Part of Speech (POS) tag such as noun, verb, or adjective. Penn TreeBank (http://www.anc.org/oanc/penn.html) is used for marking, with labels, such as, JJ for an adjective, NN for singular noun, CD for a cardinal number. The POS tags are mandatory for extracting compound words [29], such as 'software development'.

The compound word extraction uses a set of rules. The rules are designed using knowledge of words construction from English dictionaries and in-depth analysis of scientific and technical English texts. The rules were validated by English literature experts, few sample rules are shown in Table 2 with explanations, and detailed rules available in Ref. [51].

Table 2. Compound words identification and extraction rules.

Rules	Description
$\vee a, a \in N$	every compound words have noun
$\vee a, a \in N, if succeed(a, a) \rightarrow join(a, a)$	A noun succeeded by a noun, terms are joined
$\vee a,b, where a \in N \wedge b \in A,$ $if succeed(a, b) \rightarrow term(a) \wedge drop(a)$	A noun succeeded by an adjective, the noun term is saved and adjective is compared with next token

Besides the identification of compound words, POS tags also help in identifying cardinals, such as *'1' or 'three'*. The identification of cardinals has an impact on text search results, e.g., Java with two years of experience. Here, two is a cardinal that represents the level experience for the skill java. Identification of a cardinal helps to match the experience mentioned in the job description with a user query/profile. After marking the text with POS tags, compound words, cardinals, and entity extraction are next in the pipeline.

Entity extraction is a process of extracting important information from unstructured text, such as places, organizations, names, or money. Figure 4 shows domain-specific entities for e-recruitment from a job description, such as expertise level, skills, and job requirements.

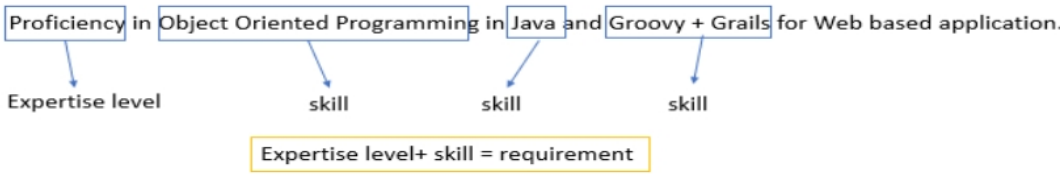

Figure 4. Basic and contextual entities extracted from a job description using SAJ.

Besides extracting the basic entities, Figure 4 shows the extraction of contextual entities, such as job requirements. The extraction of contextual entities is based on the occurrence of basic entities in some predefined order, for example the occurrence of expertise level and skill in a particular context forms a requirement.

The domain dictionary is utilized in the extraction of domain-specific entities. The domain dictionary contains patterns/action rules for the extraction of entities from text. These patterns/action rules are developed using JAPE (Java Annotation Pattern Engine) (https://gate.ac.uk/sale/tao/splitch8.html) grammar. JAPE grammar uses features for building pattern/action rules. The feature set contains aspects, such as POS tags, dictionary of words, and simple pattern/action rules. The JAPE rules also incorporate priorities for determining the execution order. The execution order affects the input/output for subsequent pattern/action rules. Example of pattern/action rules for entities are mentioned in Table 1. Section 4 discusses entity extraction along with JAPE rule examples in more detail.

3.4. Context Builder

The extracted entities are forwarded to the context builder and enrichment module in parallel. The context builder creates relationships (both hierarchical and associative) among extracted entities using a job description ontology as shown in Figure 5. The job description ontology is designed using job posting schema (https://schema.org/JobPosting) from schema.org and job description domain studies from various existing job portals discussed above. The ontology schema concepts and relationships were evaluated by HR domain experts for validating the domain coverage of job descriptions. The details of the Job Description Ontology are discussed in [52].

The job description ontology provides a schema for structuring and building the context of the extracted entities as shown in Figure 6. The core schema classes are *Job Description*, *Job Title*,

Requirements, Education, Career Level, and Job Type. Some of the core properties are the *job description, requirements, job type, education, and job title*. The ontology not only defines hierarchical relationships but also define associative relationships such as skos:altlabel, owl:sameAs and others. Figure 6 represents the requirements of a job description in an ontological model along with all its semantics. A relationship is created between a skill and an expertise level in the requirement. The relationships are not automatically extracted from the job description text rather the job description ontology already defines these relationships. The context builder uses entity types, such as skill, job requirement, expertise level, career level and others for identification of relationships.

For example, *S1* is an instance of the *Skills* class that is an intermediary node to connect *Skill* instance *Object Oriented Programming* with *Expertise Level* instance *Proficiency*. The intermediary node *S1* is then connected to *R1*, which is an instance of the *Requirement* class, connected to a *Job Description* instance *JD1*.

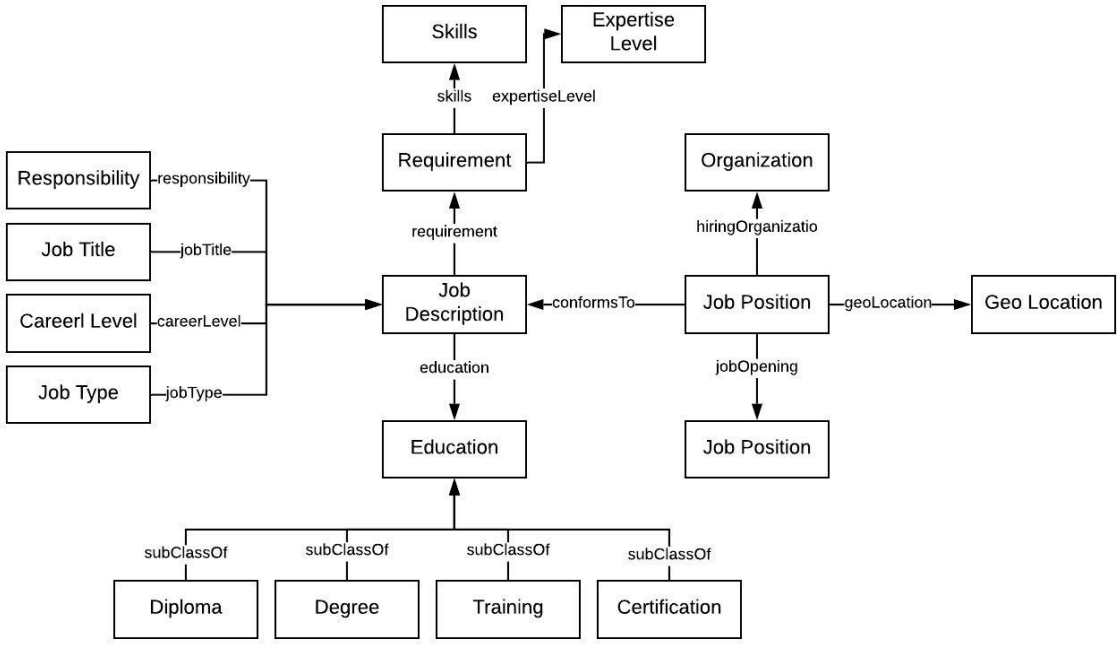

Figure 5. The proposed job description ontology with classes and properties used in SAJ.

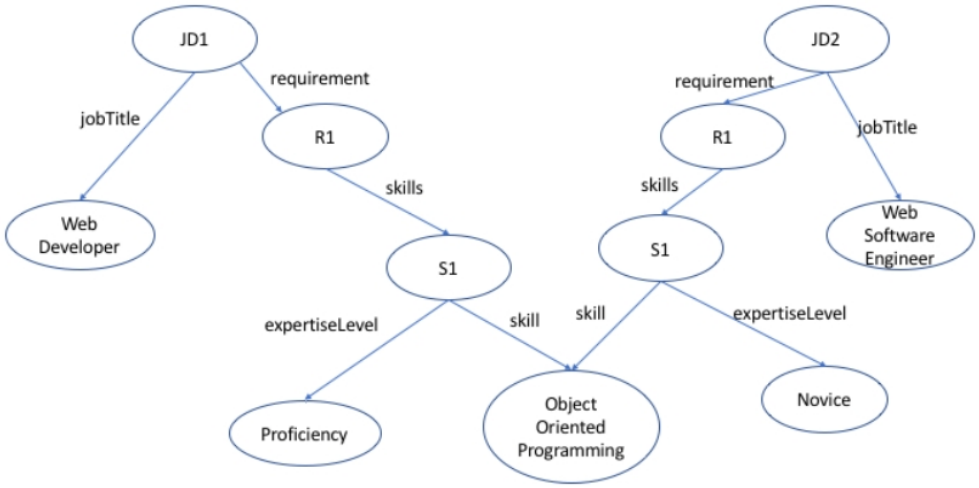

Figure 6. Graph structure showing entities and connections between the extracted entities to build a context in SAJ.

3.5. Enrichment

As mentioned before, entities extracted are also forwarded to the enrichment process of SAJ. Enrichment is a process that adds additional knowledge to existing information. The addition of information will result in improved search results [53] and job-profile matching. The current process achieves enrichment in two possible ways: (1) inter-document connections and (2) addition of knowledge from Linked Open Data. *Inter-document connection* is the result of storing data as a graph structure in the knowledge base. Figure 6 shows the concept of *object oriented programming* linked to two job descriptions with different job titles. A search that needs to find all jobs that have *object oriented programming* as a requirement will get precise results. The other method is to enrich information via Linked Open Data. During enrichment with Linked Open Data, additional terms are fetched with language conditions, for example lang=en. The reason for specifying the language is to get only English terms, not those in other languages—the current system does not have multilingual support. The similarity score is computed using cosine similarity via DISCO API between the entity extracted from the job description and that retrieved from LOD.

3.6. Knowledge Base

The knowledge base is responsible for storing the data that have been forwarded from the context builder and enrichment process after integration. The knowledge base stores information as a graph structure using the job description ontology. Currently, GraphDB (http://graphdb.ontotext.com/) is being used as knowledge base to store information. The current snapshot of the knowledge base as shown in Figure 6 visualizes two job descriptions *JD1* and *JD2*. Both job descriptions have the same *Object Oriented Programming* as a skill requirement where their expertise levels are different. The graph structure representation of the knowledge base will now connect the same instance of *Object Oriented Programming* to all *Skill* instances with different expertise levels. This structure of the knowledge base is more resourceful when exploring a query in a graph such as *find all jobs which have the requirement of object-oriented programming*.

The purpose of Section 3 has been to describe SAJ precisely. The primary discussion has been on overall information flow in SAJ. Discussion on SAJ was critical for better understanding. The primary focus of this work is on the extraction and enrichment of entities, which is discussed next in detail.

4. Extraction and Enrichment in SAJ

The extraction and enrichment of entities from a job description is a non-trivial and challenging task because similar information can be described with various ways descriptions, e.g., a requirement for a skill *java* with two or more years of experience is represented with various wordings in Table 3. This type of variation in text makes it a challenge to extract information with minimal information loss. Moreover, entity enrichment using Linked Open Data will help in resolving the contextual disambiguation of information. For example, *Java* is an island of Indonesia (https://en.wikipedia.org/wiki/Java), but, in the current context, it is a programming language mentioned as a skill. The extraction and enrichment process assists in catering for all such challenges in e-recruitment techniques.

Table 3. Sample text to show the wording variation in a job description.

1. 2+ years of experience required in Java.
2. Must have worked at least 2 years in Java development.
3. Experience of 2+ years is required in Java development.

This process consists of various steps to handle challenges in extracting information from job descriptions.

4.1. Basic Information Extraction

The entities, such as *job title*, *location*, *career level*, and *organization* are basic entities for a job description. Table 4 shows each of these entities with examples.

Table 4. Entities along with examples from a job description in SAJ.

Entity	Example
Job Title	Java Software Engineer
Location	St. Louis. MO
Career Level	Mid-Level
Organization	Google, Inc.

A job must have a job title as a mandatory entity, but others are optional. The extraction of basic information is carried out by hybrid approach using heuristics and rules from the dictionary. In the case of a job title, when more than one job title is detected, then the first line heuristic as mentioned before in Table 1 is applied. The position of the job title plays a vital role. Another aspect that requires much deliberation is the use of a special character in job title, thus creating an issue with incorrect boundary detection's of job title entities.

4.2. Requirements Extraction

Requirements are contextual entities that define the essential skills or capabilities that an employer seeks in a potential candidate. For example, the *Job Requirements* segment in Figure 1 shows the requirements in the job description. The extraction of requirements is vital due to its significance for both employers and candidates. Requirements are not just basic entities; instead, they are an association of basic entities in a specific context, e.g., a skill *Java* has various expertise levels, such as *novice and proficient*. Here, skill and its expertise level make a single requirement as they occur in a specific context as shown in Figure 4. In this process, there are two main steps: (1) identification of the requirement boundary, and (2) identification of entities.

The requirement boundary is the start and end of a requirement. Table 5 shows a sample rule that marks the boundary for a requirement. The rule uses POS tags in combination with words in a sentence to mark a boundary for the requirement.

Table 5. Sample rule for boundary detection of requirement using Java Annotation Pattern Engine (JAPE) in SAJ.

Rule	Description

`Rule:requirementboundarymarker`
`Priority: 100`
`{Lookup.majorType==Req_BeginKeywords}`
`({SpaceToken})[0,2]`
`({Token.category==IN}| {Token.category==TO}|`
`{Token.category==VBG}| {Token.category==VB}|`
`{Token.category==VBZ}|{Token.category==DT})?`
`((({SpaceToken})[0,3] ({Token.kind==word,`
`!Lookup.majorType==Req_NotAfterKeywords}`
`|{Token.kind==symbol}| {Token.kind==number}|`
`{Token.kind==punctuation,!Token.string==‘‘.’’}) )+)`
`: req --> :req.Requirement =`
`{rule = ‘‘requirementboundarymarker’’}`
	The rule mark the boundary of the requirement. It detects the token categories as Part of Speech (POS). The tokens are either verbs (VBZ, VBG, VB), determiners or prepositions. Besides the POS requirement, the keyword placement in the sentence is verified.

Once the requirement boundary is identified, the next step is to identify the actual requirement. An important aspect of the rule is setting its priority. Priorities are set in rules using the JAPE inherited

property *Priority*. The main purpose of setting the priority is to define the execution order of rules. The result obtained from one rule is input to the next rule. A higher value of *priority* defines a higher order of execution of the rule. The rule shown in Table 5 has *priority* set to 100 meaning; this is the first rule that will be executed and will make a start and end boundary for a requirement.

The contextual entity requirements are extracted from a job description using pattern/action rules defined in the dictionary. The rules identify entities from unstructured text that constitute the requirements for a job description. Consider the requirement, *Proficiency in Object Oriented Programming in Java and Groovy+Grails for Web-based application*. The rule in Table 5 will mark the boundary of the requirement, *Proficiency in Object Oriented Programming in Java*, as it is a high priority rule with value 100. Table 6 shows a pattern/action rule with priority 50—this extracts skills, such as Java, and will be executed after the rule with priority 100.

Table 6. Sample rule for job extraction of requirement using JAPE in SAJ.

Rule	Description

`Phase: requirementSubParts`
`Input: RequirementsBeg Token`
`RequirementsNot RequirementsMid RequirementsEnd`
`Skill Split ToolsAndTechnology OperatingSystem`
`Database Course TechnicalLanguage Protocol`
`ExpertiseLevel MandatoryConditionTrue`
`MandatoryConditionFalse ExpDuration`
`Options: control = appelt`
`Rule:requirementSubPartsStart`
`Priority: 50`
`{RequirementsBeg} ((((){Token})* | {Skill}`
`| {ToolsAndTechnology} | {OperatingSystem}`
`| {Database} {Course} | {TechnicalLanguage}|`
`{Protocol} |{ExpertiseLevel}|`
`{MandatoryConditionTrue} |`
`{MandatoryConditionFalse}|`
`{ExpDuration} |{ExpertiseLevel}))+{Split} )`
`:req --> :req.Requirements =`
`{rule =‘‘requirementSubPartsStart’’}`
	The rule uses various dictionaries ToolAndTechnology, OperatingSystem, Database, Course, TechnicalLanguage and others to detect the entities. Besides these entities the rules also detects the experience duration and expertise level. The rule is dynamic, i.e., placement of these entities in sentence will not affect the rule.

The dictionary defines the domain knowledge for requirement identification. The sample rule shown in Table 6 has a priority of 50 and uses various lists, such as *skill, database, course, and technical knowledge* for the extraction of requirements. A sentence is dropped if it does not satisfy any rule.

4.2.1. Responsibilities' Extraction

Responsibilities are the duties that an employee performs during their stay in an organization as shown in Figure 1. It is a non-mandatory text segment of a job description. Sometimes, responsibilities are described along with the job requirements, and are defined using similar entities, such as must have a *knowledge of AWS cloud* and will *manage AWS cloud*.

In the example, the first statement is a job requirement and the second statement is the job responsibility. It is difficult to draw a clear line of distinction between job responsibilities and job requirements. After detailed analysis and experimentation on real-world data-sets, SAJ was successful in separating job requirements from job responsibilities.

Table 7 shows the rule for detection of responsibilities from job descriptions. The sample rule extracts responsibility using domain background knowledge from a dictionary and morphological sentence structure. The rule has a low priority of 10. It uses the *Responsibilty_BeginKeywords* dictionary along with POS tags to identify the responsibility boundaries.

Table 7. Sample rule for job description responsibility detection using JAPE in SAJ.

Rule	Description
Phase: Responsibility Input: Lookup Token SpaceToken Options: control = appelt debug=true Rule:keywordResponsibility Priority: 10 {Lookup.majorType== Responsibilty_BeginKeywords} ({SpaceToken})[0,2] ({Token.category==IN}| {Token.category==TO} | {Token.category==VBG}| {Token.category==VB} | {Token.category==DT} | {Token.category==NN}| {Token.category==NNS})? ((({SpaceToken})[0,3] ({Token.kind==word, !Lookup.majorType==Res_NotAfterKeywords} | {Token.kind==symbol}|{Token.kind==number}| {Token.kind==punctuation,!Token.string==‘‘.’’}))*) :req --> :req.Responsibility = {rule = ‘‘keywordResponsibility’’}	The rule marks the boundary of the responsibility. It detects the token categories as POS. The tokens are either verbs (VBZ, VBG, VB), determiners or prepositions. Besides the POS requirement, keyword placement in the sentence is verified.

4.2.2. Education Extraction

Education defines the mandatory or minimal qualification required for the job. Education has four categories: degree, diploma, training, and certification. Besides identifying education, it is also classified into one of the predefined categories as aforementioned. This division is useful during the job matching.

Figure 7 shows an education requirement, BS/MS in Computer Science. Table 8 shows a rule for extracting the educational requirement degree. The sample rule identifies educational entities that have the token “in”. A degree dictionary is used with the POS tag to obtain the correct educational requirement identification.

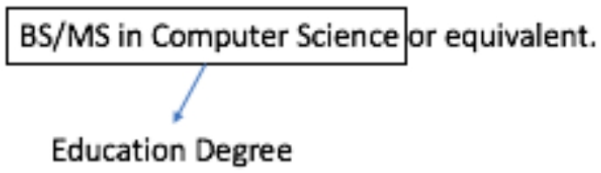

Figure 7. Sample of educational requirements in a job description.

Table 8. A sample rule for education degree extraction using JAPE in SAJ.

Rule	Description
Rule:degreeextractioninfull Priority: 40 ({Degree} {Token.string==‘‘in’’}({Token.category==NNP, !Lookup.majorType==date})+ {Token.string==‘‘and’’} ({Token.category==NNP,!Degree})+):Degree --> :Degree.FullDegree ={rule=‘‘degreeextractioninfull’’}	The rule extracts educational requirement categorized as Degree. Its use POS tag (NNP) and degree dictionaries. The rule also verifies the existence of token “in”.

4.3. Entities Enrichment

Enrichment is the process of adding additional knowledge to existing entities. The enrichment of job description entities will help with increasing the search space and better job-profile matching. The enrichment process gets its input from the entity extraction process as a list of entities. The enrichment process only processes skill entities currently. The basic aim of processing skills is to get all alternate forms, e.g., Object-Oriented Programming is also referred to OOP. The enrichment will help SAJ in identifying Object Oriented Programming and OOP as the same skill. The process achieves this by using Linked Open Data as shown in Figure 8. The main aim to use Linked Open Data for enrichment is to have up-to-date information related to the terms that are being enriched. The enrichment process via Linked Open Data will not suffer from the traditional data staleness problem. The open source community (httpsmanagethem://opensource.com/resources/what-open-source) is responsible for updating the LOD data.

The enrichment process gets additional labels from LOD based on the properties rdfs:label, rdfs:altLabel and a condition of lang=en. The rdfs:label, rdfs:altLable and lang=en filter are standard Web Ontology Language (https://www.w3.org/OWL/) properties.The data are fetched in real time when we transform unstructured text into machine-understandable structured format. No real-time query is sent to LOD when data are shown to the user.

The similarity is computed among the additional labels fetched from LOD and entities extracted from a job description. If the number of returned entities from LOD is less then five, then all returned entities are stored, but, if the number exceeds 5, then the similarity is calculated using the Cosine Similarity [54] between the terms as shown in Equation (1):

$$cos(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{||\boldsymbol{x}|| \cdot ||\boldsymbol{y}||}. \quad (1)$$

DISCO API [55] facilitates the calculation of cosine similarity. In addition to cosine similarity, a distributional similarity is also calculated using DISCO API. DISCO API allows for calculating the semantic similarity between arbitrary words and phrases. DISCO API calculates the similarities using the Wikipedia (https://www.wikipedia.org/) data set. The data-set used is from April 2013 (https://www.linguatools.de/disco/disco\protect\discretionary{\char\hyphenchar\font}{}{}download_en.html) and the API version used is DISCO API v3.0. The data-set used is of type SIM, i.e., word similarities are computed. The entities are stored in the knowledge base using skos:altLabel. SKOS is a prefix that stands for Simple Knowledge Organization System and altLabel is is used as a standard to represent alternate labels in an ontology schema.

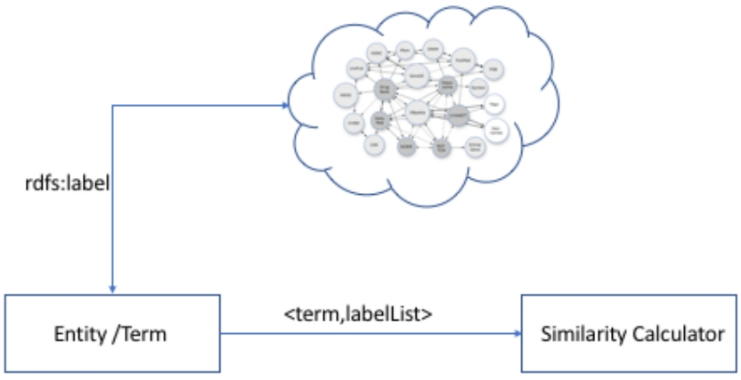

Figure 8. Entities enrichment process usingLinked Open Data (LOD) in SAJ.

5. Evaluation

The rationale for the evaluation of the proposed system derives from its main objectives, i.e., to have a system that extracts context-aware information and stores them in the knowledge base.

The information extracted by the SAJ should have minimal information loss, a larger search space and adhere to Linked Open Data principles. The current evaluation considers all these aspects.

5.1. Data-Set Acquisition

At the current moment, no gold standard data-set exists for job descriptions. The proposed e-recruitment system SAJ was evaluated on a self-collected data-set of 860 job descriptions from 2013 from various e-recruitment systems and a community mailing list. Details of the sources along with the statistics of collected job descriptions from each source are shown in Table 9. The data from Indeed and DBWorld was collected via an automated crawler. Indeed, it provides REST services to get data whereas DBWorld data was crawled. The data from Personforce (Islamabad, Pakistan) was obtained as an industrial partner.

Table 9. Statistics of jobs collected from various e-recruitment systems.

Source	Descriptions
Personforce.com (https://www.personforce.com/)	101
DBWorld (https://research.cs.wisc.edu/dbworld/browse.html)	139
Indeed.com (https://www.indeed.com.pk/?r=us)	620
Total	860

The collected job descriptions belong to multiple categories as shown in Table 10. These categories range from information technology through management to health care. The job descriptions were collected at random and then placed in these predefined categories. The random selection was to ensure that the data-set is not biased but rather contains jobs from multiple domains and disciplines.

Table 10. Statistics of job descriptions in various job categories collected randomly.

Job Category	Count
Engineering and Technical Services	55
Business Operations	20
Computer and Information Technology	125
Internet	73
Project Management	85
Health-care and Safety	9
Arts, Design and Entertainment	26
Sales and Marketing	38
Office Support and Administrative	203
Architecture and Engineering	10
Construction and Production	9
Customer Care	21
Management and Executive	22
Financial Services	9
Government and Policy	6
Post-doctoral	45
Research and Teaching	66
Others	38
Total	860

The collected data-set was evaluated by Human Resource (HR) experts who had more than five years of experience working in the area of recruitment and staffing. All entities and relationships were identified in the job descriptions. The primary entities selected after discussion with HR experts for evaluation were job title, job responsibilities, job requirements, job category and education level. These selected entities have a pivotal role in the job description(s). The results of the entity extraction from job descriptions were compared with the manually verified data from HR experts.

5.2. Evaluation Metrics

The evaluation was performed using the standard metrics of recall, precision and f-measure as shown in Equations (2)–(4) as well as an error analysis:

$$Recall = \frac{relevant - jobs \cap retrieved - jobs}{relevant - job}, \tag{2}$$

$$Precision = \frac{relevant - jobs \cap retrieved - jobs}{retrieved - jobs}, \tag{3}$$

$$F - 1 = \frac{2 \cdot precision \cdot recall}{precision + recall}. \tag{4}$$

Besides evaluating the recall, precision and f-1, overall system accuracy and error rate is also calculated, as shown in Equations (5) and (6):

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}, \tag{5}$$

$$ErrorRate = 1 - Accuracy. \tag{6}$$

5.3. Evaluation Results

Table 11 shows the results for the entities' extraction process. The table shows the recall, precision and f-measure values of various entity types. These values are computed by comparison with the gold standard data-set manually verified by HR experts. Education has the highest recall, i.e., 100%, whereas job title has the highest precision 100%. Overall job title had the highest f-1 value of 95.60%. This table shows only the proposed system evaluation results against the gold standard. In the next section, a comparison with other systems is presented.

Table 11. Results of job description evaluation based on entity types in SAJ.

No.	Entity Type	Precision	Recall	F-1
1	Requirements	90.5	87.90	88.76
2	Responsibilities	76.14	75.00	75.76
3	Education	38	100	93.60
4	Job Title	100	90.67	95.00
5	Job Category	79.24	97.67	87.50

Besides making a comparison on the basis of standard parameters of precision, recall and f-1, an accuracy vs. error comparison was also computed for SAJ, to investigate how well the SAJ performs. From the graph in Figure 9, it is quite evident that education has a low error rate of Zero. The 100% accuracy is only due to low variation in education entity. The system overall has accuracy of 94% and an error rate of 6%.

Besides measuring extraction precision and system accuracy, an evaluation was performed by retrieving the job from the knowledge base to evaluate how well extraction was performed and context was built. The main aim of this evaluation was to show that, when a user searches for jobs from the job description knowledge base, they will get highly precise and accurate results meaning that user confidence in the system will increase. For each query, jobs were manually counted and then applied to job descriptions in the knowledge base. The queries were written in the SPARQL language. The results are shown in Table 12 and clearly show that SAJ performed well in terms of extraction and context building for the job description in the knowledge base.

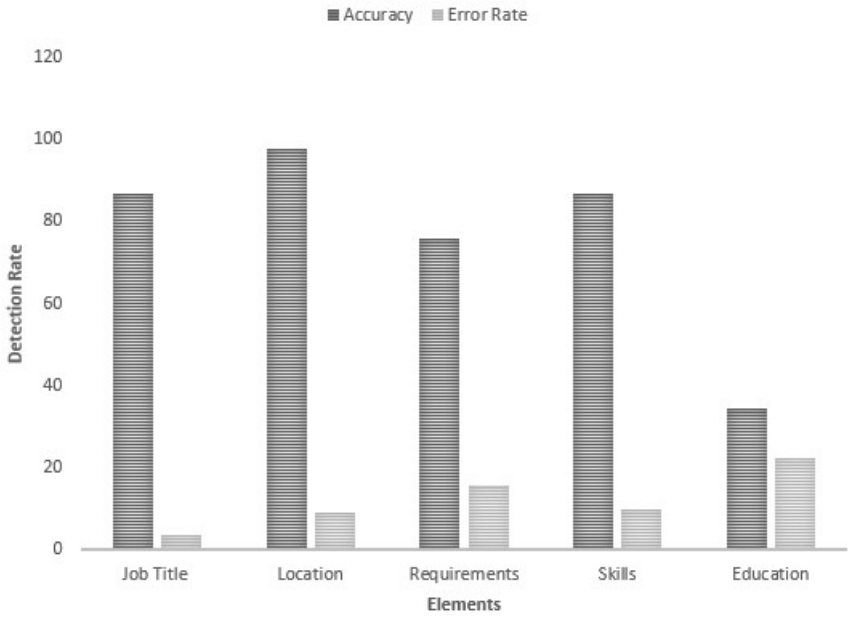

Figure 9. Evaluation comparison of Accuracy vs. Error for SAJ.

Table 12. Job description user retrieval summary.

Category	Manual	System Retrieved
Job Titles	25	25
Requirements	33	33
Career Level	45	45

5.4. Comparative Analysis

This section presents the result comparison among SAJ, OpenCalais (http://www.opencalais.com/about) and Alchemy API (http://www.alchemyapi.com/about-us). Both OpenCalais and Alchemy API are industry leaders in information extraction.

All systems were able to extract job titles. Figure 10 shows a comparison of the three systems for job titles' extraction. The comparison parameters are precision, recall, and f-1.

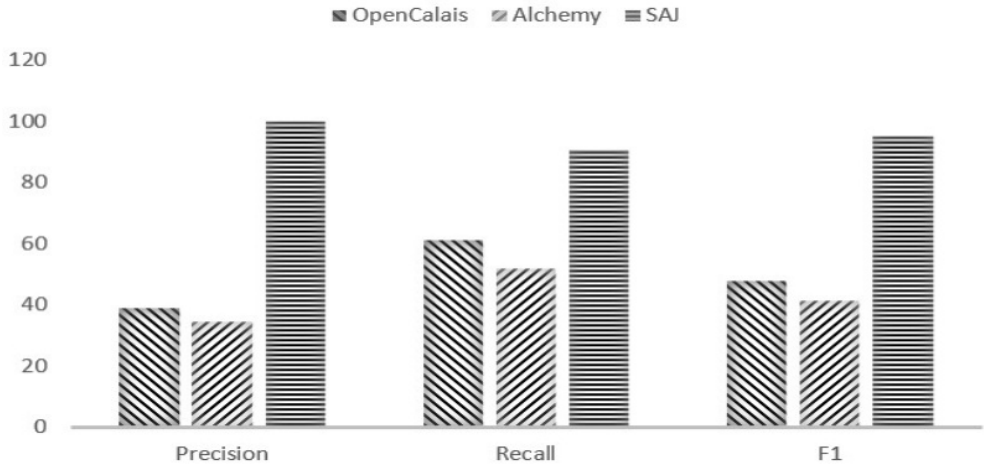

Figure 10. Comparative analysis among SAJ, Alchamy API and OpenCalais for Job Titles.

From the graph, it is evident that SAJ performs well when compared to OpenCalais and Alchemy API. SAJ has achieved an overall precision of 98.1% compared to OpenCalais 39% and Alchemy API

34.32%. The other entity that OpenCalais was able to extract and Alchemy API was not able to extract was the job requirements. The graph in Figure 11 shows a comparative analysis of the requirement entity between SAJ and OpenCalais.

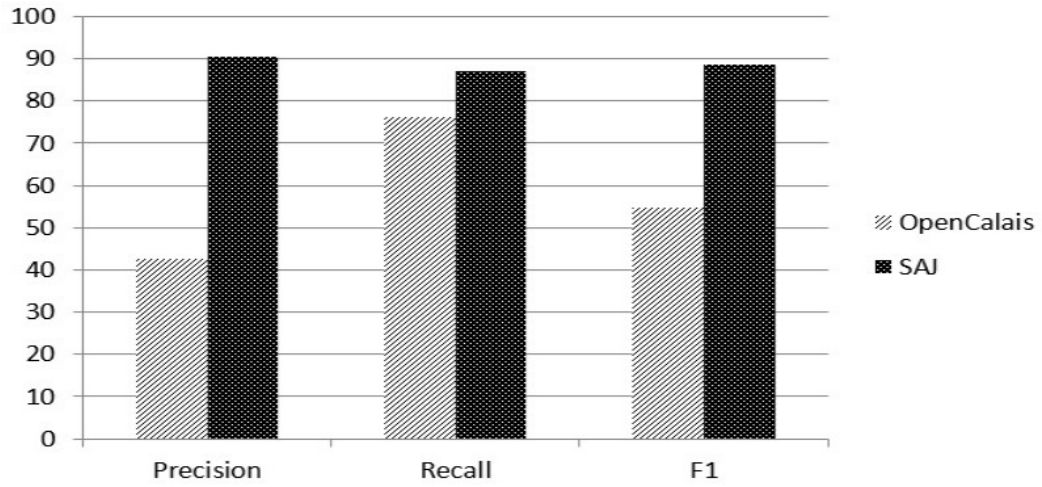

Figure 11. Comparative analysis among SAJ and OpenCalais for requirements.

From the graph in Figure 11, it is evident that SAJ has a much higher precision 90.5% than OpenCalais 42.78%. OpenCalais has a recall of 76.1%, whereas SAJ has higher recall of 87.09%. No comparison exists for education, responsibilities and job category as OpenCalais and Alchemy API were not able to extract those entities.

6. Conclusions and Future Work

In this research, SAJ extracts context-aware information from job descriptions by exploiting Linked Open Data, job description domain ontology, and domain-specific dictionaries. SAJ enriches and builds context between extracted entities to minimize the information loss in the extraction process. It combines various processes together to achieve context-aware information extraction and enrichment from job description in e-recruitment. SAJ segments the text into predefined categories using a self-generated dictionary. Natural Language Processing (NLP) and a dictionary help in identification of entities. The extracted entities are enriched using Linked Open Data, and job context is built using a job description domain ontology. The enriched and context-aware information is stored in the knowledge base built using Linked Open Data principles. The evaluation has been performed on a data-set of 860 jobs, verified by HR experts. The initial assessment was conducted by a comparison between manually verified data and system extracted entities. SAJ achieved an overall f-1 of 87.83 %. In comparison with other techniques, OpenCalais and alchemy API, SAJ performed the best. OpenCalais was able to extract job titles and job requirements while Alchemy API was only able to extract job titles. SAJ can facilitate searching and retrieval, scoring and ranking of job candidates. In the future, the plan is to extend this work for automatic dictionary learning. This will enhance the dictionary by the addition of new rules learned from the text.

Author Contributions: Conceptualization, M.N.A.A. and K.L.; Methodology, M.N.A.A.; Software, M.N.A.A.; Validation, S.K. and K.L.; Writing—Original Draft Preparation, M.N.A.A.; Writing—Review & Editing, M.N.A.A., S.K., K.L. and A.M.K.; Visualization, M.N.A.A.; Supervision, S.K. and K.L.; Project Administration, S.K. and K.L.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Is Your Recruitment Process Costing You Time, Money and Good Candidates? Available online: https://ckscience.co.uk/is-your-recruitment-process-costing-you-time-money-and-good-candidates/ (accessed on 1 January 2019).
2. Owoseni, A.T.; Olabode, O.; Ojokoh, B. Enhanced E-recruitment using Semantic Retrieval of Modeled Serialized Documents. *Int. J. Math. Sci. Comput.* **2017**, 1–16. [CrossRef]
3. Valle, E.D.; Cerizza, D.; Celino, I.; Estublier, J.; Vega, G.; Kerrigan, M.; Ramírez, J.; Villazón-Terrazas, B.; Guarrera, P.; Zhao, G.; et al. SEEMP: An Semantic Interoperability Infrastructure for e-Government Services in the Employment Sector. In Proceedings of the 4th European Semantic Web Conference, Innsbruck, Austria, 3–7 June 2007; pp. 220–234.10.1007/978-3-540-72667-8_17. [CrossRef]
4. Silvello, G.; Bordea, G.; Ferro, N.; Buitelaar, P.; Bogers, T. Semantic representation and enrichment of information retrieval experimental data. *Int. J. Digit. Libr.* **2017**, *18*, 145–172, doi:10.1007/s00799-016-0172-8. [CrossRef]
5. Romero, G.C.; Esteban, M.P.E.; Such, M.M. Semantic Enrichment on Cultural Heritage collections: A case study using geographic information. In Proceedings of the 2nd International Conference on Digital Access to Textual Cultural Heritage, DATeCH 2017, Göttingen, Germany, 1–2 June 2017; pp. 169–174, doi:10.1145/3078081.3078090. [CrossRef]
6. Introduction to the Principles of Linked Open Data. Available online: https://programminghistorian.org/en/lessons/intro-to-linked-data (accessed on 1 December 2018).
7. Sa, C.D.; Ratner, A.; Ré, C.; Shin, J.; Wang, F.; Wu, S.; Zhang, C. Incremental knowledge base construction using DeepDive. *VLDB J.* **2017**, *26*, 81–105, doi:10.1007/s00778-016-0437-2. [CrossRef]
8. Gregory, M.L.; McGrath, L.; Bell, E.B.; O'Hara, K.; Domico, K. Domain Independent Knowledge Base Population from Structured and Unstructured Data Sources. In Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference, Palm Beach, FL, USA, 18–20 May 2011.
9. Buttinger, C.; Pröll, B.; Palkoska, J.; Retschitzegger, W.; Schauer, M.; Immler, R. JobOlize - Headhunting by Information Extraction in the Era of Web 2.0. In Proceedings of the 7th International Workshop on Web-Oriented Software Technologies (IWWOST'2008) in conjunction with the 8th International Conference on Web Engineering (ICWE'2008) Yorktown Heights, New York, NY, USA, 14 July 2008.
10. Karkaletsis, V.; Fragkou, P.; Petasis, G.; Iosif, E. Ontology Based Information Extraction from Text. In *Knowledge-Driven Multimedia Information Extraction and Ontology Evolution - Bridging the Semantic Gap*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 89–109, doi:10.1007/978-3-642-20795-2_4.
11. Jayram, T.S.; Krishnamurthy, R.; Raghavan, S.; Vaithyanathan, S.; Zhu, H. Avatar Information Extraction System. *IEEE Data Eng. Bull.* **2006**, *29*, 40–48.
12. Bijalwan, V.; Kumar, V.; Kumari, P.; Pascual, J. KNN based machine learning approach for text and document mining. *Int. J. Database Theory Appl.* **2014**, *7*, 61–70. [CrossRef]
13. Vicient, C.; Sánchez, D.; Moreno, A. Ontology-Based Feature Extraction. In Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, Lyon, France, 22–27 August 2011; pp. 189–192, doi:10.1109/WI-IAT.2011.199. [CrossRef]
14. Poria, S.; Cambria, E.; Ku, L.; Gui, C.; Gelbukh, A.F. A Rule-Based Approach to Aspect Extraction from Product Reviews. In Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP), Dublin, Ireland, 24 August 2014; pp. 28–37, doi:10.3115/v1/W14-5905. [CrossRef]
15. Rocktäschel, T.; Singh, S.; Riedel, S. Injecting Logical Background Knowledge into Embeddings for Relation Extraction. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, CO, USA, 31 May–5 June 2015; pp. 1119–1129.
16. Mykowiecka, A.; Marciniak, M.; Kupsc, A. Rule-based information extraction from patients' clinical data. *J. Biomed. Inform.* **2009**, *42*, 923–936. doi:10.1016/j.jbi.2009.07.007. [CrossRef] [PubMed]
17. Ramakrishnan, C.; Mendes, P.N.; Wang, S.; Sheth, A.P. Unsupervised Discovery of Compound Entities for Relationship Extraction. In *Knowledge Engineering: Practice and Patterns*; Gangemi, A., Euzenat, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 146–155.

18. Zhang, N.R. *Hidden Markov Models for Information Extraction*; Technical Report; Stanford Natural Language Processing Group: Stanford, CA, USA; 2001.
19. Lafferty, J.D.; McCallum, A.; Pereira, F.C.N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, 28 June–1 July 2001; pp. 282–289.
20. Kiryakov, A.; Popov, B.; Terziev, I.; Manov, D.; Ognyanoff, D. Semantic annotation, indexing, and retrieval. *J. Web Sem.* **2004**, *2*, 49–79, doi:10.1016/j.websem.2004.07.005. [CrossRef]
21. Popov, B.; Kiryakov, A.; Kirilov, A.; Manov, D.; Ognyanoff, D.; Goranov, M. KIM—Semantic Annotation Platform. In Proceedings of the The Semantic Web—ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, 20–23 October 2003; pp. 834–849, doi:10.1007/978-3-540-39718-2_53. [CrossRef]
22. Muller, H.M.; Kenny, E.E.; Sternberg, P.W. Textpresso: An ontology-based information retrieval and extraction system for biological literature. *PLoS Biol.* **2004**, *2*, e309. [CrossRef] [PubMed]
23. Ali, F.; Kim, E.K.; Kim, Y. Type-2 fuzzy ontology-based opinion mining and information extraction: A proposal to automate the hotel reservation system. *Appl. Intell.* **2015**, *42*, 481–500, doi:10.1007/s10489-014-0609-y. [CrossRef]
24. Saggion, H.; Funk, A.; Maynard, D.; Bontcheva, K. Ontology-Based Information Extraction for Business Intelligence. In Proceedings of the 6th International Semantic Web Conference, Busan, Korea, 11–15 November 2007; pp. 843–856, doi:10.1007/978-3-540-76298-0_61. [CrossRef]
25. Geibel, P.; Trautwein, M.; Erdur, H.; Zimmermann, L.; Jegzentis, K.; Bengner, M.; Nolte, C.H.; Tolxdorff, T. Ontology-Based Information Extraction: Identifying Eligible Patients for Clinical Trials in Neurology. *J. Data Semant.* **2015**, *4*, 133–147, doi:10.1007/s13740-014-0037-5. [CrossRef]
26. Vijayarajan, V.; Dinakaran, M.; Tejaswin, P.; Lohani, M. A generic framework for ontology-based information retrieval and image retrieval in web data. *Hum.-Centric Comput. Inf. Sci.* **2016**, *6*, 18. [CrossRef]
27. Al-Yahya, M.M.; Aldhubayi, L.; Al-Malak, S. A Pattern-Based Approach to Semantic Relation Extraction Using a Seed Ontology. In Proceedings of the 2014 IEEE International Conference on Semantic Computing, Newport Beach, CA, USA, 16–18 June 2014; pp. 96–99, doi:10.1109/ICSC.2014.42. [CrossRef]
28. Vicient, C.; Sánchez, D.; Moreno, A. An automatic approach for ontology-based feature extraction from heterogeneous textualresources. *Eng. Appl. AI* **2013**, *26*, 1092–1106, doi:10.1016/j.engappai.2012.08.002. [CrossRef]
29. Ahmed, N.; Khan, S.; Latif, K.; Masood, A. Extracting Semantic Annotation and their Correlation with Document. In Proceedings of the 4th International Conference on Emerging Technologies, Rawalpindi, Pakistan, 18–19 October 2008, pp.32–37.
30. Mayfield, J.; McNamee, P.; Harmon, C.; Finin, T.; Lawrie, D. KELVIN: Extracting knowledge from large text collections. In Proceedings of the 2014 AAAI Fall Symposium, Arlington, VA, USA, 13–15 November 2014; pp. 34–41.
31. Zhang, C.; Shin, J.; Ré, C.; Cafarella, M.J.; Niu, F. Extracting Databases from Dark Data with DeepDive. In Proceedings of the 2016 International Conference on Management of Data, SIGMOD, San Francisco, CA, USA, 26 June–1 July 2016; pp. 847–859, doi:10.1145/2882903.2904442. [CrossRef]
32. Cafarella, M.J.; Ilyas, I.F.; Kornacker, M.; Kraska, T.; Ré, C. Dark Data: Are we solving the right problems? In Proceedings of the 32nd IEEE International Conference on Data Engineering, ICDE, Helsinki, Finland, 16–20 May 2016; pp. 1444–1445, doi:10.1109/ICDE.2016.7498366. [CrossRef]
33. Richardson, M.; Domingos, P.M. Markov logic networks. *Mach. Learn.* **2006**, *62*, 107–136, doi:10.1007/s10994-006-5833-1. [CrossRef]
34. Gao, N.; Dredze, M.; Oard, D.W. Knowledge Base Population for Organization Mentions in Email. In Proceedings of the 5th Workshop on Automated Knowledge Base Construction, AKBC@NAACL-HLT, San Diego, CA, USA, 17 June 2016; pp. 24–28.
35. Weichselbraun, A.; Gindl, S.; Scharl, A. Enriching semantic knowledge bases for opinion mining in big data applications. *Knowl.-Based Syst.* **2014**, *69*, 78–85, doi:10.1016/j.knosys.2014.04.039. [CrossRef] [PubMed]

36. Bertin, M.; Atanassova, I. Semantic Enrichment of Scientific Publications and Metadata: Citation Analysis Through Contextual and Cognitive Analysis. *D-Lib Mag.* **2012**, *18*, doi:10.1045/july2012-bertin. [CrossRef]
37. Sun, H.; Ma, H.; Yih, W.; Tsai, C.; Liu, J.; Chang, M. Open Domain Question Answering via Semantic Enrichment. In Proceedings of the 24th International Conference on World Wide Web, WWW, Florence, Italy, 18–22 May 2015; pp. 1045–1055, doi:10.1145/2736277.2741651. [CrossRef]
38. Hsueh, H.Y.; Chen, C.N.; Huang, K.F. Generating metadata from web documents: a systematic approach. *Hum.-Centric Comput. Inf. Sci.* **2013**, *3*, 7. [CrossRef]
39. Russo, V. Semantic Web: Metadata, Linked Data, Open Data. *Sci. Philos.* **2017**, *3*, 37–46.
40. Yamaguchi, A.; Kozaki, K.; Lenz, K.; Yamamoto, Y.; Masuya, H.; Kobayashi, N. Data Acquisition by Traversing Class-Class Relationships over the Linked Open Data. In Proceedings of the ISWC 2016 Posters & Demonstrations Track Co-Located with 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, 19 October 2016.
41. Koho, M.; Hyvönen, E.; Heino, E.; Tuominen, J.; Leskinen, P.; Mäkelä, E. Linked Death–representing, publishing, and using Second World War death records as Linked Open Data. In *The Semantic Web: ESWC 2017 Satellite Events, European Semantic Web Conference, Anissaras, Greece, 29 May 2016*; Springer: Cham, Switzerland, 2016; pp. 3–14.
42. Kamdar, M.R.; Musen, M.A. PhLeGrA: Graph Analytics in Pharmacology over the Web of Life Sciences Linked Open Data. In Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, 3–7 April 2017; pp. 321–329, doi:10.1145/3038912.3052692. [CrossRef]
43. Wei, K.; Huang, J.; Fu, S. A survey of e-commerce recommender systems. In Proceedings of the 2007 International Conference on Service Systems and Service Management, Chengdu, China, 9–11 June 2007; pp. 1–5.
44. Roman, D.; Kopecký, J.; Vitvar, T.; Domingue, J.; Fensel, D. WSMO-Lite and hRESTS: Lightweight semantic annotations for Web services and RESTful APIs. *J. Web Sem.* **2015**, *31*, 39–58, doi:10.1016/j.websem.2014.11.006. [CrossRef]
45. Sharifi, O.; Bayram, Z. A Critical Evaluation of Web Service Modeling Ontology and Web Service Modeling Language. In *Computer and Information Sciences, Proceedings of the International Symposium on Computer and Information Sciences, Krakow, Poland, 27–28 October 2016*; Springer: Cham, Switzerland; pp. 97–105, doi:10.1007/978-3-319-47217-1_11.
46. Rekha, R.; Syamili, C. Ontology Engineering Methodologies: An Analytical Study. 2017. Available online: https://pdfs.semanticscholar.org/abba/aec8969745162d25d3f468dc080eda289ce7.pdf (accessed on 15 March 2018).
47. Malherbe, E.; Cataldi, M.; Ballatore, A. Bringing Order to the Job Market: Efficient Job Offer Categorization in E-Recruitment. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; pp. 1101–1104, doi:10.1145/2766462.2776779. [CrossRef]
48. Sen, A.; Das, A.; Ghosh, K.; Ghosh, S. Screener: a system for extracting education related information from resumes using text based information extraction system. In Proceedings of the International Conference on Computer and Software Modeling, Cochin, India, 20–21 October 2012; Volume 54, pp. 31–35.
49. Malik, S.K.; Prakash, N.; Rizvi, S. Semantic annotation framework for intelligent information retrieval using KIM architecture. *Int. J. Web Semant. Technol.* **2010**, *1*, 12–26. [CrossRef]
50. Cunningham, H.; Cunningham, H.; Maynard, D.; Maynard, D.; Tablan, V.; Tablan, V. JAPE: A Java Annotation Patterns Engine, 1999. Available online: https://www.researchgate.net/publication/2495768_JAPE_a_Java_Annotation_Patterns_Engine (accessed on 16 April 2010).
51. Awan, M.N.A. Extraction and Generation of Semantic Annotations from Digital Documents. Master's Thesis, NUST School of Electrical Engineering & Computer Science, Islamabad, India, 2009.
52. Ahmed, N.; Khan, S.; Latif, K. Job Description Ontology. In Proceedings of the International Conference on Frontiers of Information Technology, FIT, Islamabad, Pakistan, 19–21 December 2016; pp. 217–222, doi:10.1109/FIT.2016.047. [CrossRef]
53. Agichtein, E.; Brill, E.; Dumais, S.T. Improving web search ranking by incorporating user behavior information. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, DC, USA, 6–11 August 2006; pp. 19–26, doi:10.1145/1148170.1148177. [CrossRef]

54. Thada, V.; Jaglan, V. Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm. *Int. J. Innov. Eng. Technol.* **2013**, *2*, 202–205.
55. Kolb, P. Disco: A multilingual database of distributionally similar words. In *KONVENS 2008-Ergänzungsband: Textressourcen und Lexikalisches Wissen*; 2008; Volume 156. Available online: https://pdfs.semanticscholar.org/e280/07775ad8bd1e3ecdca3cea682eafcace011b.pdf (accessed on 20 July 2010).