2013 IEEE 37th Annual Computer Software and Applications Conference Workshops

Towards an Information Extraction System based on Ontology to Match Résumés and Jobs

[1]Duygu Çelik, [2]Aşkın Karakaş, [3]Gülşen Bal, [4]Cem Gültunca, [5]Atilla Elçi, [6]Başak Buluz, [7]Murat Can Alevli

Kariyer.net, [2]askin@kariyer.net, [3] gulsenb@kariyer.net, [4]info@cemgultunca.com.tr

Istanbul Aydin University, [1]duygucelik_sw@hotmail.com, [6]basakbuluz@gmail.com, [7]canalevli@hotmail.com

Aksaray University, [5]atilla.elci@gmail.com

***Abstract*- While Internet takes up by far the most significant part of our daily lives, finding jobs/employees on the Internet has started to play a crucial role for job seekers and employers. Online recruitment websites and human resources consultancy and recruitment companies enable job seekers to create their résumé, a brief written formal document including job seeker's basic information such as personal information, educational information, work experience and qualifications in order to find and apply for desirable jobs, whereas they enable companies to find qualified employees they are looking for. However résumés may be written in many ways that make it difficult for online recruitment companies to keep these data in their relational databases. In this study, a project that Kariyer.net (largest online recruitment website in Turkey) and TUBITAK (The Scientific and Technological Research Council of Turkey) have been jointly working is proposed. In this mentioned project, a system enables free structured format of résumés to transform into an ontological structure model. The produced system based on ontological structure model and called Ontology based Résumé Parser (ORP) will be tested on a number of Turkish and English résumés. The proposed system will be kept in Semantic Web approach that provides companies to find expert finding in an efficient way.**

***Keywords*- Ontology, Semantic Web, Information Extraction, Résumé, Curriculum Vitae.**

I. INTRODUCTION

Internet has been used not only for business and communicative purposes but also for daily life activities such as shopping, online banking, reading online news etc. Finding a job on the Internet is quite popular since it enables job seekers to see all the job vacancies and catch easily career opportunities.

In addition, companies/institutions are able to hire efficiently qualified employees with the help of online recruitment websites. Storage of millions of résumés in free-structured format in relational databases of the companies is highly time consuming and requires a great deal of human effort.

Semantic Web [1] is an extension of World Wide Web that aims to enable computers to discover, search, infer and collect Web's information without human effort. Semantic Web allows efficient way of representing data on the World Wide Web. Thus, the information may be comprehended by machines. Ontology is a term that refers to define and make connections between information.

Web Ontology Language (OWL) is a standard ontology language from World Wide Consortium (W3C) that processes and instantiates Ontologies [2]. In Semantic Web Technology, computers are able to comprehend web information and make inferences of them; and this very proposed system is based on Semantic Web technology. Several related studies designed by using ontologies for English language are described below:

U. Bojārs and J. G. Breslin proposed **Résumé RDF** ontology for describing a résumé semantically. The purpose of the system is to reveal the structure of *'authority finder'*. Résumé RDF may describe information such as skills, work experience, academic experience and features in a semantic way [3]. Another similar study is **Description of a Career (DOAC)**, a vocabulary suggested by R. A. Parada in order to describe résumés [4]. In DOAC concepts about information, features, capabilities or skills of people were depicted yet not implemented by ontology. DOAC is implemented by vocabulary. By the comparison of these two systems, it is shown that Résumé RDF gives more query result than DOAC. Studies mentioned above are based on English language and they prefer to use exact matching with the sentence/word found from document with a pre-defined résumé vocabulary. Some other studies are referenced in [5-11] which parsing algorithms are based on syntactic-based matching algorithm instead of the semantic matching. In other words, the system makes reference that ***ITU*** is an abbreviation and refers to *Istanbul Technical University,* whereas it can't make inference the semantics of *'It is a university',* in other words *'ITU is a university'*. However, the

978-0-7695-4987-3/13 $26.00 © 2013 IEEE
DOI 10.1109/COMPSACW.2013.60

system proposed is called *Ontology Résumé Parser (ORP)* and will work both for résumés written in English and Turkish languages. The system aims to parse information from a résumé such as general information, personal information, education information, work experience, qualifications, projects , certificates, references, other information etc. and to analyze its data and infer new concepts from the written ontological rules with existing data. The system makes inference with the predefined semantic rules based on the resume knowledge that makes it differ substantially from other studies. Furthermore, there is no Ontology Knowledge Base (OKB) for Turkish language. In the literature, ORP will be the first to work for résumés written in Turkish or English language. The proposed system may be used online recruitment websites in order to provide fast and accurate information extraction from job seeker's résumés.

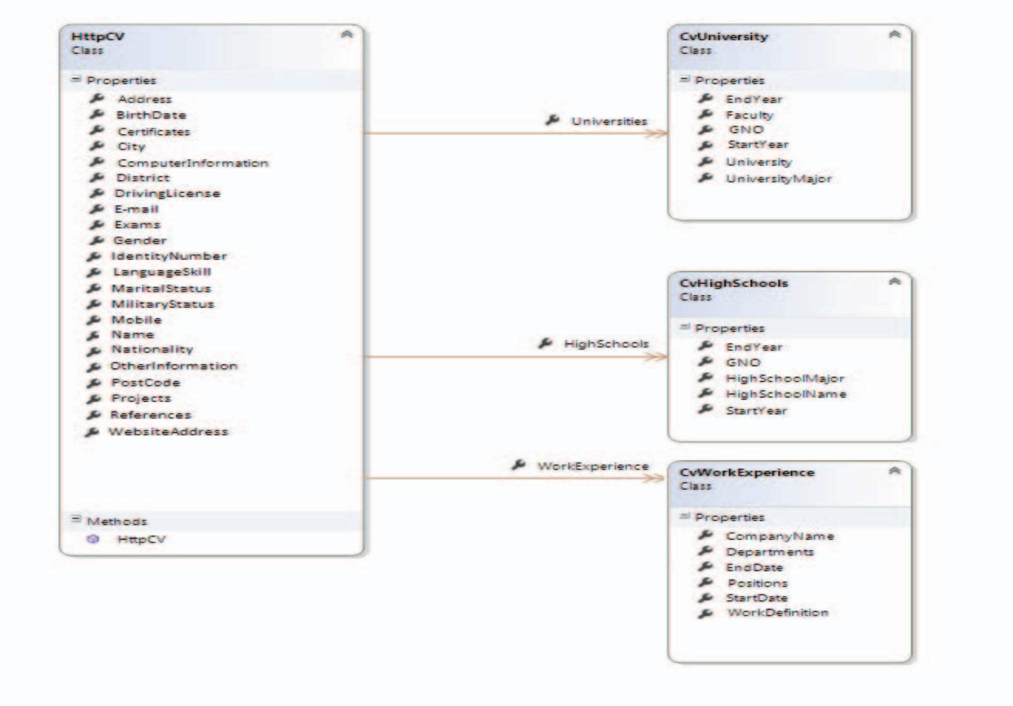

Fig 1.Class Diagram of the system

The creation of résumé ontology is programmed under Protégé[1] editor [12]. The proposed system will be developed under Java platform Netbeans[2] that associates to OKB with the OWL API[3] (Ontology Parser). Through OWL API, proposed system saves information which is obtained from resume to Resume Ontology. Additionally, defined system rules can be run on Resume Ontology and the system can generate new facts from existing ones.

Following sections are Ontology Knowledge Base (OKB), implementation, tests&results and conclusion. In the OKB section, the detailed description of OKB, its working mechanism and example of ontology are mentioned. In the implementation part, how résumé parsing processes are conducted and how its results are saved on the ontology database are explained.

II. ONTOLOGY KNOWLEDGE BASE

Ontology Knowledge Base of an ORP has a Résumé Ontology (RO) including classes, properties and concepts with relationships. The system has its own semantic rules which aim to infer new concepts/facts from existing ones that is mentioned next section. RO enables to represent parts of a résumé in a semantic way. RO has many parts such as personal information, general information, work experience, educational information, skills, certifications, exams, other information, personal/professional references etc. Ontological structure of a résumé and semantic web rules are created by Protégé editor[1] [12]. The proposed system has a general résumé format that may cover thousands of free-structured résumé formats that are kept in the Kariyer.net company database[4].

As mentioned above a résumé has many parts and there are sub-parts of specific parts such as Personal

[1] http://protege.stanford.edu/

[2] https://netbeans.org/

[3] http://owlapi.sourceforge.net/

[4] www.kariyer.net

Information can have sub-parts such as *Birth Date, Birth Place, Driving License, Military Status, and Marital Status*.

General Information consists of individual's information such as *Name, Email, Home Address, and Cell Phone Number.*

Educational information keeps data of *university* and *college names*, *college* and *university departments, university faculty* and *start-end date* of *university/college* information of a person.

Work experience indicates data of current and past work experiences. Its sub parts consist of *Company Name*, *Position, Department, Start-End Date* of a work and *Work Description*.

Additionally, skills of a person have sub-parts of *Computer Skills*, *Language Skills* and *Certificates*. The class diagram of a résumé can be seen in Figure1.

A Word document is parsed based on defined system rules and extract information from a résumé and save them to HttpCV class. As seen in Figure 1, HttpCV class has many instances of other classes. It shows combination of a whole résumé parts and also includes lists that hold information related to educational information and work experience. After information extractions from a résumé and save obtained information to HttpCV classes are done, HttpCV Class values are saved to ontology. The ontology that has specific classes, properties and individuals to load HttpCV values is called Résumé Ontology (RO). A small portion of a class structure can be seen in Table 1.

Résumé ontology starts with *'owl: Thing'* class. Under this, the 'CV' class is defined as a subclass. Fourteen different subclasses are further defined under the CV class. These classes are: *CVEducationExperience*, *CVSegments*, *CVWorkExperience*, *Exam*, *GeneralInformation*, *OtherInfo*, *Reference*, *PersonalInformation*, *Skills*, *Projects*, *Organization*, *Nationality*, *MartialStatus* and *Location*.

Additionally, some classes have sub-classes in order to keep detailed information. For instance, *Location*

TABLE I. LABELS OF CLASS IN RÉSUMÉ ONTOLOGY

```
<Ontology xmlns=http://www.w3.org/2002/07/owl#
<OntologyIRI="http://www.....CVOntology.owl">
<Declaration><Class IRI="#Organization "/></Declaration>
<Declaration><Class IRI="#CVEducationExperience"/></Declaration>
<Declaration><Class IRI="#CVSegments"/> </Declaration>
<Declaration><Class IRI="#CVWorkExperience"/> </Declaration>
<Declaration> <Class IRI="#CVGeneralInformation"/> </Declaration>
<Declaration><Class IRI="#CVPersonalInformation"/> </Declaration>
<Declaration><Class IRI="#Exam "/> </Declaration>
<Declaration><Class IRI="#OtherInfo "/> </Declaration>
<Declaration> <Class IRI="#Projects "/> </Declaration>
<Declaration><Class IRI="#Location "/> </Declaration>
<Declaration><Class IRI="#Skills "/> </Declaration>
<Declaration><Class IRI="#References"/> </Declaration>
<Declaration> <Class IRI="#Nationality "/> </Declaration>
<Declaration><Class IRI="#MartialStatus"/> </Declaration>
<Declaration> <Class IRI="#CVHighSchools "/> </Declaration>
<Declaration><Class IRI="#CVUniversity"/> </Declaration>
<Declaration><Class IRI="#City"/> </Declaration>
<Declaration><Class IRI="#District "/> </Declaration>
<Declaration> <Class IRI="#CV "/> </Declaration>
<Declaration><Class IRI="#Gender"/> </Declaration>
<Declaration><Class IRI="#Certificates"/> </Declaration>
```

TABLE II. LABELS OF DATA AND OBJECT PROPERTY OF RÉSUMÉ ONTOLOGY

```
<Ontology xmlns=http://www.w3.org/2002/07/owl#
<OntologyIRI="http://www..../CVOntology.owl">
<Declaration>
        <ObjectPropertyIRI="#hasPersonalInformation"/>
</Declaration>
<Declaration>
        <ObjectProperty IRI="#hasGeneralInformation "/>
</Declaration>
<Declaration>
        <ObjectProperty IRI="#hasWorkExperience"/>
</Declaration>
<Declaration>
        <ObjectProperty IRI="# hasReferences"/>
</Declaration>
<Declaration>
        <ObjectProperty IRI="# hasLanguage"/>
</Declaration>
<Declaration>
        <ObjectProperty IRI="# hasExam "/>
</Declaration>
<Declaration>
        <ObjectProperty IRI="# hasGender"/>
</Declaration>
<Declaration>
        <ObjectProperty IRI="#hasOtherInfo"/>
</Declaration>
<Declaration>
        <ObjectProperty IRI="#hasNationality"/>
</Declaration>
<Declaration>
        <ObjectProperty IRI="# hasMilitary "/>
</Declaration>
<Declaration>
        <DataProperty IRI="#hasAddress "/>
</Declaration>
<Declaration>
        <DataProperty IRI="#hasBirthDate"/>
</Declaration>
<Declaration>
        <DataProperty IRI="#hasRésuméID"/>
</Declaration>
<Declaration>
        <DataProperty IRI="#hasWorkYear "/>
</Declaration>
<Declaration>
        <DataProperty IRI="#hasEmail"/><
</Declaration>
<Declaration>
        <DataProperty IRI="#hasMobile"/>
</Declaration>
<Declaration>
        <DataProperty IRI="#hasWebPage "/>
</Declaration>
<Declaration>
        <DataProperty IRI="#Name "/>
/Declaration>
<Declaration>
        <DataProperty IRI="#hasPostCode"/>
</Declaration>
```

class has sub-classes such as *City* and *District* in order to indicate definite location of a person. Ontology is created in a way that all of the possible information of résumé can be kept in Résumé Ontology. *CVEducationInformation* class keeps data of *University* and *High School* from which job seeker graduated. *CVWorkExperience* class keeps data of current/previous works such as work position, work department, end/start of work, and company/institute name and work description. Exam class represents taken exams, *GeneralInformation* class keeps data of general information of person such as person's name, address information, website, mobile phone number, email address and post code.

OtherInfo class represents information of additional activities, hobbies. *PersonalInformation* class keeps data of birth date, birth place, and identity number, driving license, gender, and nationality, military and marital status of person. *Reference* class keeps data of given references of person. *Skills* class keeps data of skills of person such as computer skills, language skills and certifications.

Project class keeps data regarding the information of projects achieved; *Organization* class not only keeps data of the organizations of all types and sizes such as companies, institutes, universities, high schools, hospitals etc. but also keeps data of all possible departments and job positions. *Nationality* class includes data of all the nationalities; *MaritalStatus* class keeps data of marital status types such as 'single' or 'married'. *Location* class keeps data of all the cities and districts. Moreover, *Segments/titles* plays an important role to define specific parts of résumé as well. *CVSegments* class keeps track of information regarding found titles in résumé. If the system finds specific titles on résumé, it becomes easier to parse information beneath title and operate certain rules based on title. As a result, class structure keeps data of all information obtained from résumé and declared information imported from database tables on Kariyer.net. In order to define relations between classes, system defines object and data properties as seen in Table 2. In this project, common concepts of résumé are defined then *Object* and *Data* properties in OWL language are created. *Data Property* uses literals to avoid uncertainty of data such as *hasAddress*, *hasBirthDate*, *hasEmail* etc. whereas *Object Property* demonstrates relation between concepts such as *hasPersonalInformation*, *hasGeneralInformation*, *hasWorkExperience* etc. Detailed example is shown in Table 2. After that, an individual of CV is generated based on the upper RO (Table 1 and Table 2) and then create individual of *PersonalInformation* (Table 3). Afterwards, connect individuals with object properties. As seen in Table 3, storage of information about university such as university name, department name and faculty information is saved as object property. Information regarding *StartDate* and *EndDate* is saved as data property and kept as literals (Table 3 line 35-42). When a required information extraction process of an individual is conducted, parsed information from the résumé is saved to in the RO format in to a separate file. OWL API II[5] has been used to extract or add information from Résumé Ontology. At this stage, it is possible to insert, extract information from ontology and add/get data or object properties from ontology with the OWL API for the focused individual resume file.

TABLE III. CREATION AND DEFINITION OF EDUCATION EXPERIENCE AND UNIVERSITY

```
1   <ClassAssertion>
2   <Class IRI="#CV "/>
3   <NamedIndividual IRI="#Peter_James"/>
4   </ClassAssertion>
5   <ClassAssertion>
6   <Class IRI="#CVEducationExperience "/>
7   <NamedIndividual IRI="#EE "/>
8   </ClassAssertion>
9   <ObjectPropertyAssertion>
10  <ObjectProperty IRI="#hasEducationExperience"/>
11  <NamedIndividual IRI="#Peter_ James"/>
12  <NamedIndividual IRI="#EE"/>>
13  <ObjectPropertyAssertion>
14  <ObjectProperty IRI="#hasUniversity"/>
15  <NamedIndividual IRI="#EE"/>
16  <NamedIndividual IRI="#Uni"/>
17  <ObjectPropertyAssertion>
18  <ObjectProperty IRI="#hasUniversityName"/>
19  <NamedIndividual IRI="#Uni"/>
20  <NamedIndividual IRI="#Columbia_University "/>
21  </ObjectPropertyAssertion>
22  <ObjectPropertyAssertion>
23  <ObjectProperty IRI="#hasUniversityDepartmentName"/>
24  <NamedIndividual IRI="#Uni"/>
25  <NamedIndividual IRI="#Computer_Science "/>
26  </ObjectPropertyAssertion>
27  <ObjectPropertyAssertion>
28  <ObjectProperty IRI="#hasUniversityFacultyName"/>
29  <NamedIndividual IRI="#Uni"/>
30  <NamedIndividual IRI="#Faculty of Engineering and
31  Architecture "/>
32  </ObjectPropertyAssertion>
33  <DataPropertyAssertion>
34  <DataPropertyAssertion>
35  <DataProperty IRI="#StartDate"/>
36  <NamedIndividual IRI="# Columbia_University "/>
37  <Literal datatypeIRI="&xsd;string">2008</Literal>
38  </DataPropertyAssertion>
39  <DataProperty IRI="#EndDate"/>
40  <NamedIndividual IRI="# Columbia_University "/>
41  <Literal datatypeIRI="&xsd;string">2012</Literal>
42  </DataPropertyAssertion>
```

In order to run ***Semantic Web Rule Language (SWRL)***[6] rules and produce new facts from existing facts, a reasoner is required to work with OWL API and JAVA. The system uses Pellet reasoner [13] which is standard and advance guard reasoning services for OWL ontologies. During the inferencing

[5] http://owlapi.sourceforge.net/

[6] *http://www.w3.org/Submission/SWRL/*

task, the system uses Pellet reasoner through some predefined SWRL rules. The rules are semantically defined and discussed by using the for both linguistics rules of English and Turkish languages in our other study [14]. The study presents some particular SWRL rules based on the RO format that are applied on predefined criterias for a particular job position to find an expert for the particular field. For instance, a human resource staff is looking for a *'Java Supervisor'* expert as a candidate for a company's job position and searches for a person who has university degree, Java knowledge and more than three years' experience. The concepts of a person and a university can be captured from OWL classes called '*Person'* and '*University'*. The length of experience, degree and computer skill conditions can be expressed as *hasExperienceYR*, *hasdegreefrom* and *hasComputerSkill* object properties form of this OWL formed résumé. This rule could be written in SWRL form as:

Person(?p) ^ University(?u) ^ hasComputerSkill (?p,?JAVA) ^ hasdegreefrom(?p,?u)

^ hasExperienceYR(?p, ?yr) ^ greaterThan(?yr, 3) → hasTitle(?p, JavaSupervisor)

TABLE IV. PSEUDOCODE OF INFORMATION EXTRACTION SYSTEM FROM RÉSUMÉS

```
1   Convert Docx formatted résumé to HTML format with
2   DuoDimension Tool.
3
4   Remove HTML tags to get salt text of document.
5   Remove unnecessary tags
6   Replace '\n','\r' , '\t' characters with " " (Space) character
7   Replace <br>,<li> tags with '\r'
8   Replace <p>,<div>,<tr> tags with '\r\r'
9   Convert contiguous space characters more than two to one
10
11  Apply SentenceEnd Algorithm
12  IF end of a sentence is found
13  THEN add character '\r'
14  END
15
16  Call PutTag function to find sentences<stc> and
17  paragraphs<prf>
18  Split all sentences with " " (Space) character and save them to a
19  list
20
21  Compare every word with Resume Ontology concepts values
22  IF word matches in Acronyms table values in Ontology
23  THEN take value of acronym representation of word
24  END
25
26  Save obtained paragraphs <prf> and sentences <stc> to XML file
27  Add attributes for each sentence and paragraphs and set them
28  to 0
29
30  WHILE all words are not checked.
31  IF word matches with any table values
32  THEN write category name and increase its attribute value by 1
33  END
34  Based on process rules defined in RULE table apply algorithms
35  Get next word to process
36  Go to step 20.
37  END
38
39  Find highest attribute value of each paragraph
40  Extract information based on defined rules in RULES table in
41  Ontology
```

III. IMPLEMENTATION

The proposed system performs information extraction from free-structured docx formatted résumés. Table 4 denotes a part of the pseudocode of the information extraction system from a résumé.

As it poses difficulty to parse and understand tags of word format, system converts word formatted résumés to HTML format. Besides unnecesary tags in HTML occur, thus system removes these tags such as <img>,<script>, <table>, <tfoot>, <thead>,<u> etc. Furthermore, replace '\n', '\r', '\t' tags with space character in order to eliminate line breaks and tabs. Afterwards, replace <br>,<li> tags with '\r' and <p>,<div>,<tr> tags with '\r\r' and then remove more than two space characters which are contigous and put one space character instead of them. Then, system applies *SentenceEnd* algorithm, if sentence end is found '\r' tag is added. In order to understand paragraphs and sentences from html document, these tag replacements are used. *PutTag* function searches whole document if there is '\r', then puts <stc> tag at the end of word which means it is a sentence. Likewise, if there is '\r\r' tag, then puts <prf> tag at the end of word which means it is a paragraph. Then, system split every paragraph's every sentence by the ' ' character to find all words and save them to a list. Before scoring process begins, all the words are compared with Acronyms concepts in ontology to find extended meaning of word. System saves obtained paragraphs and sentences with their tags (prf, stc). Thereafter certain tags to figure out paragraphs' meaning such as it is a personal information(pi), education information(ei), name(na), general information(ge), skills(sk), other information(ot), references(re), title id, (ti) and work experience(we) are added and all tag values are set to 0. Besides, sentences to bold(b), italic(i) and underline(u) tags to understand sentence's feature are added and set to 'false'. Category tag to sentence to figure out sentence's category are added as well. In order to understand each sentence's type, system starts to score process based on Rule collection in the resume ontology. First, system compares every word with the every related concepts or individuals in ontology knowledgebase and discovers the word's category/concept and write it to sentence's '*category'* tag. Secondly, system checks if there are any titles in document by looking into 'Titles' concept in ontology. If any title is found, write defined title category in Title to category of a sentence and increment attribute value with increasing value defined in Title concept. Next, based on category value in rule base, the system continues scoring process like apply N-Gram Algorithm[7] which creates

[7] http://en.wikipedia.org/wiki/N-gram

different combinations of words or apply Jaro-Wrinkler Algorithm[8] [15-16] to eliminate misspelling errors or use Regex to check mail addresses or mobile, dates etc. Subsequent to conduction of scoring process, system calculates highest attribute value of each paragraph and then based on defined rules in Rule base in OKB, system conducts information extraction. After information extraction is conducted, it is saved first HttpCV format and then saved to résumé ontology to make new inferences for expert finding step.

IV. TESTS & RESULTS

The proposed system is tested with 250 turkish resumes and acquire average 80% of information extraction from resumes. The calculation of information extraction is presented below:

$$\frac{Extracted\ Word\ Numbers}{Total\ Word\ Numbers} \times 100$$

System run time is 5 second averagely.To prevent wrong information extraction, the system applies defined rules on specific parts. To illustrate, if a paragraph is an education experience paragraph then only education experience information extraction rules will be applied not work experience rules. The system defines some main parts/sections which have to be defined in a resume such as Work Experience, Education Information, General Information etc. Based on these sections, the system calculates percentage completeness of a resume to indicate how many parts are completed averagely.

V. CONCLUSION

The proposed ontology-driven information extraction system that is called Ontology-based Résumé Parser (ORP) will be operated on few millions of English language and Turkish language résumés to convert them to an ontological format. The overall objective of the proposed ORP system is based on concept matching task and ontological rules for English and Turkish résumés which provide for semantic analysis of data and parse related information such as experiences, features, business and education information from a résumé. The system contains its own Ontology Knowledge Base (OKB). To conclude, the working system mechanism, the OKB, matching steps, transferring of plaintext résumé into ontology form and the inference mechanism though SWRL rule base are discussed in this article. Further details of above mentioned SWRL based inferencing mechanism of the system is discussed separate study [14].

[8] http://en.wikipedia.org/wiki/Jaro-Winkler_distance

REFERENCES

[1] Berners-Lee T, Hendler J, Lassila O. The Semantic Web. Sci Am, 2001, 284: 34–43.

[2] OWL Web Ontology Language Overview, W3C Recommendation, Online: http://www.w3.org/TR/owl-features/last accessed on April 2009.

[3] Bojars U, Breslin J G. RésuméRDF: Expressing Skill Information on the Semantic Web. The 1st International Workshop on ExpertFinder. Berlin, Germany, January 2007.

[4] Parada R A. DOAC Vocabulary Specification. http://wiki.foaf-project.org/w/DOAC. July 2006, last visited: March 2013.

[5] Karamatlı E. and Akyokuş S. (2010), Resume Information Extraction with Named Entity Clustering based on Relationships, INISTA 2010, Kayseri-Turkey.

[6] Sovren Résumé / CV Parser, http://www. sovren.com / last visited: April 12, 2012.

[7] ALEX Résumé Parsing, http://www.hireability.com/ ALEX / last visited: April 12, 2012.

[8] Résumé Grabber Suite http://www.egrabber. com / résumégrabbersuite / last visited: April 12, 2012.

[9] Daxtra CVX, http://www.daxtra.com/, last visited: April 12, 2012.

[10] Piskorski J, Kowalkiewicz M, Kaczmarek T. Information Extraction from CV. Information Retrieval and Filtering, pp. 185-192, 2005.

[11] Yu K, Guan G, Zhou M. Résumé information extraction with cascaded hybrid model. ACL 2005, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Morristown, NJ, USA, pp. 499-506, 2005.

[12] Protégé OWL Ontology Editor, http://protege.stanford.edu/, last visited: March 19, 2013.

[13] Sirin E, Parsia B. PELLET. An OWL DL Reasoner. In: International Workshop on Description Logics (DL2004), Whistler, 2004.

[14] Celik D, Elci A. An Ontology-Based Information Extraction Approach for Résumés. ICPCA/SWS 2012: The 7th International Pervasive Computing and the Networked World, İstanbul, Turkey, October 26-28.

[15] Winkler W E. The state of record linkage and current research problems In: Technical report, U.S. Census Bureau, Washington, D.C, 1999.

[16] Winkler W E, Thibaudeau Y. An application of the Fellegi-Sunter model of record linkage to the 1990 U.S. decennial census. In: Technical report, U.S. Bureau of the Census, Washington, D.C: Statistical Research Report Series RR91/09, 1991.