Ontology Extraction and Semantic Ranking of Unambiguous Requirements

Subha R.[1], Palaniswami S. [2]

[1]Sri Krishna College of Technology, Coimbatore , 641 042, India
[2] Government College of Engineering , Bodinayakanur 625 528, Tamil Nadu, India .
kris.subha@gmail.com

Abstract: This paper describes a new method for ontology based standardization of concepts in a domain. In Requirements engineering, abstraction of the concepts and the entities in a domain is significant as most of the software fail due to incorrectly elicited requirements. In this paper, we introduce a framework for requirements engineering that applies Semantic Ranking and significant terms extraction in a domain. This work aims to identify and present concepts and their relationships as domain specific ontologies of particular significance. The framework is build to detect and eliminate ambiguities. Semantic Graph is constructed using semantic relatedness between two ontologies which is computed based on highest value path connecting any pair of the terms. Based on the nodes of the graph and their significance scores, both single as well as multi word terms can be extracted from the domain documents. A reference document of ontologies that will help requirement analyst to create SRS and will be useful in the design is created.
[Subha R. ,Palaniswami S. **Ontology Extraction and Semantic Ranking of Unambiguous Requirements.** *Life Sci J* 2013;10(2):131-138].(ISSN: 1097-8135). http://www.lifesciencesite.com. 21

Keywords: Content based retrieval, Information Retrieval, Semantics, Software Engineering

Introduction

Requirements engineering is a significant phase in Software Engineering and is concerned with establishing a common understanding of the requirements to be addressed by the software product. Requirements engineering is the first phase of Software Engineering that tries to understand the needs of a system and produces a consistent, complete set of requirements in a standard format. It includes requirements elicitation, analysis and specification. At the end of the requirements engineering phase, we get a document containing the description of the system. This is called Software Requirements specifications (SRS). This document serves as a basis for the design and also as a reference throughout the software engineering life cycle (Saba et al., 2011). It consists of a set of transformations that attempt to understand the exact needs of a software-intensive system and convert the statement of the needs into a complete and unambiguous description of the requirements, a specified standard. This area includes knowledge of the requirements activities of elicitation, analysis, and specification. Requirement engineering produces one large document containing a description of what the system will do without describing how it will do. This document is known as Software Requirement Specification (SRS) and this documentation works as the foundation for the design of the software (Saba and Rehman, 2012). "An SRS is unambiguous if, and only if, every requirement stated therein has only one interpretation", as stated in IEEE Recommended Practice for Software Requirements Specifications. SRSs are usually written in natural language, often enhanced by information in other notations, such as formulae, and diagrams. An online survey of businesses requiring software, conducted at University of Trento in Italy shows that a majority of documents available for requirements analysis are provided by the user or are obtained by interviews. Moreover,

- 71.8% of these documents are written in common natural language,
- 15.9% of these documents are written in structured natural language, and
- Only 5.3% of these documents are written in formalized language

Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications.Ambiguity is an essential feature to be considered as it affects the Natural language requirements document and thereby affects the quality of the software (Berry, 2003). Many pre-processing activities are involved to carry out the ambiguity detection and classification. Earlier, the requirements gathering team was equipped with a handbook in order to remove ambiguity while preparing the requirements documents. When it comes to reading the requirements documents, they are large in amount and time consuming. The Natural Language requirements documents are error-prone. Before analyzing the requirements documents for the levels of ambiguity, the evaluation of requirements documents needs to be

considered as it involves a significant role in analyzing the document characteristics. Ontology describes the concepts and relationships that are vital in a particular domain, providing a glossary for that domain as well as an automated specification of the meaning of terms used in the vocabulary. Ontology helps in developing a shared understanding across a project. Ontology represents a consensual, shared description of the pertinent objects considered as existing in a certain area of knowledge. They constitute a special kind of software artifact conveying a certain conception of the world. This is specifically designed with the purpose of explicitly expressing the intended meaning of a set of agreed existing objects. The software engineering ontology defines common sharable software engineering knowledge including particular project information. Software engineering ontology typically provides software engineering concepts – what they are, how they are related, and can be related to one another – for representing. Ontologies range from taxonomies and classifications, database schemas, to fully axiomatized theories. In recent years, ontologies have been implemented in many business and scientific communities as a way to share, reuse and process domain knowledge. Ontologies are now central to many applications such as scientific knowledge portals, information management and integration systems, electronic commerce, and semantic web services. Semantic matching is a type of ontology matching technique that is based on ranking of ontologies to identify nodes that are semantically related. Given any two graph-like structures, like classifications, database or XML schemas and ontologies, matching is an operator that identifies those nodes in the two structures which semantically correspond to one another. Abstraction is the process of extracting the most important information from a source (to produce a condensed version for a particular user and task .The domain expert has the rich background information needed for the structuring the requirements, whereas the requirements engineer has to evaluate its relevance to the context (Hovy, 2005). Domain expertise is available to the requirements engineer as documents. These documents may be formal documents such as standards, problem descriptions or existing system specifications, or they may be less formal documents such as interview transcripts or field reports of ethnographic studies. Identifying abstractions from such documents is a tedious task for a requirements engineer.

In this paper, we propose a method that builds a reference document of ontologies that will help requirement analyst to create SRS and will be useful in the design. Such a reference document will contain the significant terms in the domain. These significant terms are extracted based on the frequency present in the domain documents. Multiple documents may be available in domain and semantically similar terms may be represented in different forms in different documents. Hence semantically tracking the terms across different documents is very essential. The document terms are identified and TF-IDF value is computed for all sensible words. Ambiguity is an inherent phenomenon in natural language and the ambiguities in the document are detected and classified. After elimination of ambiguities, each word is compared with all other words. The highly meaningful word of each word is computed. From the computed value, semantic-graph is constructed. In graph construction, semantic relatedness between two terms is identified. This relatedness between two terms can be computed based on highest value path connecting any pair of the terms. In finding highest value, the different meanings (senses) that appear between each word are found (Sultanov, 2011).From the terms and their semantic relatedness, ontologies are extracted.

The paper is organized as follows: section 2 discusses the related work, section 3 explains the methodology, section 4 deals with the experiment and results and section 5 is the concluding part of the paper.

Related Work

Ananiadou presented a domain independent method for the automatic extraction of multi-word terms, from machine-readable special language corpora (Ananiadou, 1994). The method(C-value/NC-value), combines linguistic and statistical information. The first part, C-value enhances the common statistical measure of frequency of occurrence for term extraction, making it sensitive to a particular type of multi-word terms, the nested terms. The second part, NC-value, gives: a method for the extraction of term context words (words that tend to appear with terms) and the incorporation of information from term context words to the extraction of terms.

Brigitte Orliac and Mike Dillinger reported on the development of a collocation extraction system (Orliac, 2003).Here, robust syntactic analysis is used to refine collocation extraction. Embedding the extraction system also addressed the need to provide information about the source language collocations in a system-specific form to support automatic generation of a collocation rule base for analysis and translation.

Diana Maynard and Sophia Ananiadon adopted an approach which used a variety of knowledge sources – syntactic, semantic and statistical – and attempted to both enlighten and make use of the theoretical foundations of terminology in a practical application (Maynard, 1999). The work sought to identify those parts of the context which are most relevant to the terms. Contextual weights are incorporated based on a new similarity measure, into a method for term recognition, and thereby improve the ranking of terms and enable disambiguation to be

achieved.

Ricardo Gacitua, Pete Sawyer and Vincenzo Gervasi proposed a new technique for automated abstraction identification called relevance-based abstraction identification (RAI), and evaluated its performance (Gacitua, 2011). Here, abstraction is based upon the frequency of terms present in the domain documents. This approach had not taken the semantic relatedness between the terms into account.

George Tsatsaronis, Irakilis Varlamis and Kjetil Nervag stated Semantic Rank, a graph-based ranking algorithm for keyword and sentence extraction from the text which constructs a semantic graph using implicit links, which are based on semantic relatedness between text nodes and consequently ranks nodes using different ranking algorithms (Tsatsaronis, 2010).

Chinatsu Aone and Scott William Bennett proposed "Applying Machine Learning to Anaphora Resolution" (Aone , 1996).This system uses feature vectors for pairs of an anaphor and its possible antecedent. A total of 66 features are used, and they include lexical (e.g. category), syntactic (e.g. grammatical role), semantic (e.g. semantic class), and positional (e.g. distance between anaphor and antecedent) features. Those features can be either unary features (i.e. features of either an anaphor or an antecedent such as syntactic number values) or binary features (i.e. features concerning relations between the pairs such as the positional relation between an anaphor and an antecedent).

Hui Yang, Anne de Roeck, Vincenzo Gervasi, Alistair Willis and Bashar Nuseibeh (Yang, 2011)developed architecture of an automated system to support requirements writing, by incorporating nocuous ambiguity detection into the requirements workflow. The core of such architecture comprises a classifier that automatically determines whether an instance of anaphoric ambiguity is nocuous or innocuous. The classifier is developed using instances of anaphoric ambiguity extracted from a collection of requirements documents. For each instance, a set of human judgments are used to classify. A classifier is then trained on the linguistic features of the text and the distribution of judgments to identify instances of nocuous ambiguity in new cases. Several approaches can be followed to ensure a good quality requirements document. Another approach is the linguistic analysis of a NL requirements document intended to confiscate most of the issues related to readability and ambiguity. A lot of studies dealing with the evaluation and the achievement of quality in NL requirement documents can be found in the literature and Natural Language Processing (NLP) tools have been recently applied to NL requirements documents for inspecting the consistency and completeness.

Methodology

The proposed ontology extraction framework has been designed to support abstraction identification in Requirements Engineering. It combines a number of existing natural language processing techniques with significant terms extraction to enable it to handle both single and multiword terms, ranked in order of confidence. One of the main contributions of this paper is the semantic ranking of terms and use of ambiguity detection and classification, which avoids the problems associated with semantic relatedness between the terms. The input is the domain document available in different formats. Hence they are to be pre processed. The documents after pre processing are analysed for ambiguity, then the ambiguities are classified and eliminated. The unambiguous requirements are semantically ranked and evaluated based on their significance. From these significant terms and semantic relationships, ontologies are extracted.

A. Text Pre-Processing

1) Sentence Detector

Sentence Detector can detect that a punctuation character marks the end of a sentence or not. A sentence is defined as the longest white space trimmed character sequence between two punctuation marks. The first and last sentence makes an exception to this rule. The first non whitespace character is assumed to be the beginning of a sentence, and the last non whitespace character is assumed to be a sentence end.

2) Tokenizer

The Tokenizers segment an input character sequence into tokens. Tokens are usually words, punctuation, numbers, etc. Most part-of-speech taggers, parsers and so on, work with text tokenized in this manner. It is important to ensure that tokenizer produces tokens of the type expected by later text processing components.

3) Named entity recognizer

The Name Finder can detect named entities and numbers in text. To be able to detect entities the Name Finder needs a model. The model is dependent on the language and entity type it was trained for. The Open NLP projects offer a number of pre-trained name finder models which are trained on various freely available corpora.

4) POS Tagger

The Part of Speech Tagger marks tokens with their corresponding word type based on the token itself and the context of the token. A token can have multiple pos tags depending on the token and the context. The POS Tagger uses a probability model to guess the correct pos tag out of the tag set. To limit the possible tags for a token a tag dictionary can be used which increases the tagging and runtime performance of the tagger.

5) Parser

Domain expertise is available in various reference documents .These documents are parsed to get tokens of words. From the tokens, stop words are removed .Porter's algorithm is applied to perform stemming (Porter M, 1980).

B. Ambiguity Detection

Ambiguity detection includes coreference resolution and ambiguity classification.

1) Co reference Resolution

Co reference Resolution is the process of identifying the linguistic expressions which make reference to the same entity or individual within a single document or across a collection of documents. Co reference occurs when multiple expressions in a sentence or document refer to the same entity in the world. Initially all possible references need to be extracted from the document before determining the co reference for a document. Every reference is a possible anaphor, and every reference before the anaphor in document order is a possible antecedent of the anaphor, except when the anaphor is nested. If the anaphor is a child or nested reference, then the possible antecedents must not be any reference with the same root reference as the current anaphor. (Vincent, 2002) (Soon,2001) Still, the possible antecedents can be other root references and their children that are before the anaphor in document order. The new ambiguous instance, potential pairs of co referring NPs are offered to the classifier to resolve whether the two NPs co refer or not in order to estimate the co reference relations among the possible NPs antecedent candidates. In this system, heuristics-based methods are built-in to exploit the factors that influence co reference determination. The heuristics are incorporated in terms of feature vectors and are modeled based on the Table I.

Table I. Feature Vector Description For Coreference Resolution Heuristics

FEATURE TYPE	FEATURE	DESCRIPTION
String matching	Full string matching	Y if both NPs contain the same string aftere the removal of non-informative words,else N
	Head word matching	Y if both NPs contain the same Headword,else N
	Modifier matching	Y if both NPs share the same modifier substring, else N
	Alias name	Y if one NP is the alias name of the other NP, else N
Grammatical	NP type (NP_i)	Y if NP_i is either definite NP or demonstrative NP, else N
	NP type (NP_j)	Y if NP_j is either definite NP or demonstrative NP, else N
	Proper name	Y if both NPs are proper names, else N
	Number agreement	Y if NP_i and NP_j agree in number, else N
Syntactic	PP attachment	Y if one NP is the PP attachment of the other NP, else N
	Appostive	Y if one NP is in appostive to the other NP, else N
	Syntactic role	Y if both NPs have the same syntactic role in the sentence, else N

Each instance of an anaphor is associated with a set of candidate antecedents. A pair wise comparison of the NPs is accomplished by the classifier to identify potential co reference relations among the candidate antecedents. Likewise, each NP pair is tested for co reference, and sets of co referent candidates are identified.

2) Ambiguity Classification

Anaphoric ambiguity (Dagan, 1990, Rehman and Saba,2011) occurs when the text offers two or more potential antecedent candidates either in the same sentence or in a preceding one, as in, 'The function shall build the parse tree, and then display it in a new window'. The expression to which an anaphor (Saba and Rehman,2012) refers is called its antecedent. Antecedents for personal pronoun (Brennan, 1987) anaphora are nouns or noun phrases (NPs) found elsewhere in the text, usually preceding the anaphor itself. Based on multiple human judgments of the suitable NP antecedent candidate in terms of an anaphoric ambiguity instance (Yang,2011, Rehman et al., 2013), the antecedent can be classified. A number of preference heuristics are also included to model the factors that may favor a particular interpretation. A machine learning algorithm is implemented with a set of training instances to construct a classifier. Given an anaphor and a set of possible NP antecedents, the classifier then predicts how strong the preference for each NP is, and from there, whether the ambiguity is nocuous or innocuous. The Naive Bayes classifier is used to classify the antecedents. The Naive Bayes

classifier uses the feature vectors in to classify the antecedent and the anaphoric ambiguity. Some of the features are shown in Table II.

Table II. Feature Vector Description For Antecedent Classification Heuristics

FEATURE TYPE	FEATURE	DESCRIPTION
Linguistics	Number agreement	Y if NP agree in number; N_P if NP does not agree in number but it has a person property; N if NP doesn't agree in number; UNKNOWN if the number information cannot be determined
	Definiteness	Y if NP is a definite NP; else N Non-prepositional NP Y if NP is a non-prepositional NP; else N
	Syntactic constraint	Y if NP satisfies syntactic constraint; else N
	Syntactic parallelism	Y if NP satisfies syntactic parallelism; else N
	Indicating verb	Y if NP follows one of the indicating verbs; else N
	Semantic constraint	Y if NP satisfies semantic constraint; else N
	Semantic parallelism	Y if NP satisfies semantic parallelism; else N
	Domain-specific term	Y if NP is contained in the domain-specific term list; else N
Context	Centering	Y if NP occurs in the paragraph more than twice; else N
	Section heading	Y if NP occurs in the heading of the section; else N
	Sentence recency	INTRA_S if NP occurs in the same sentence as the anaphor; else INTER_S
	Proximal	Integral value n, where n means that NP is the nth NP to the anaphor in the right-to-left order
Statistics	Local-based collocation frequency	Integral value n, where n refers to the occurrence number of the matched co-occurrence pattern containing NP in local requirements document
	BNC-based collocation frequency	Y if the matched co-occurrence pattern containing NP appears in the word list returned by the sketch engine; else N

The machine learning algorithm allots a weighted antecedent tag to the NP candidate while they are presented with a pronoun and a candidate.

C. Significance Based Extraction

The relevant terms in an unfamiliar domain may be either a single or multiword. The extraction of the multiword is a difficult process due to the structural ambiguity present in English language (Sultanov, 2011). The Semantic relatedness between the terms is also a problem in the extraction of terms. To overcome this, semantic ranking based significant terms extraction is proposed. Corpus based Frequency Profiling is applied in which terms are extracted based on their occurrence. Significant terms extraction is implemented to identify terms, i.e., sequences of tokens. The tokens are mined based on their frequency. A document D is written in natural language. K is the static source of knowledge not dependent on D. A_D is the set of terms to be extracted. The extracted term can be either a single or multi word. Significance score is measured for the terms and ranked (Yue, 2011) .

Measure of significance

The measure of significance is given by

$$\sigma : A_D \longrightarrow R$$

A_D depends on two factors, namely whether it includes all and only significant terms and whether its associated ranking corresponds to different degree of significance.

D. Semantic Ranking Algorithm

For the set of terms extracted from the document, TF-IDF value is computed and semantic-graph is constructed. Term frequency $TF_{t,d}$ is the count of number of times a particular word / term occurred in a document. Inverse document frequency IDF_t is calculated using

$$IDF_t = \log (N/DF_t)$$

where N is total number of document in the collection, DF_t is document frequency i.e. number of documents where a given term occurs. Using $TF_{t,d}$, IDF_t, TF-IDF weight for each term is calculated using

$$TF\text{-}IDFt,d = TFt,d * IDFt$$

In graph construction, semantic relatedness between two terms is identified. Semantic relatedness between two terms can be computed based on highest value path connecting any pair of the terms (Hovy, 2005)(Diaz-Aviles ,2009). In finding highest value, the different meanings (senses) that appear between each word are determined. The highest value terms obtained from the semantic relatedness are sorted in descending order. The documents with highest important word similarity are ranked as top position. This forms a

Graph structure i.e., the document which get many number of important words gets the highest priority node (Tsatsaronis, 2010). Other documents with least words are ranked next.

In case of single word relevance calculation is done with the following procedure. A significant word w in the domain document is assumed .The domain document contains a total of n_d words. The normative corpus contains n_c words.w occurs w_d times in the domain document and w_c times in the normative corpus.w_d and w_c are called the observed values of w. Based on the occurrence of w in the domain document and normative corpus the two expected values for w are determined.

In case of multiword, ranking has to be done in order of the relevance of their signified abstractions. To handle this, log-likelihood value for each word is assigned as in single word. Syntactic patterns are applied to the text to identify multiword terms. The headword is given more significance by assigning a weight factor k to words of multiword. Then the significance score is calculated and the terms are ranked.

E. Ontological Matching And Extraction

The linguistic matching algorithm V-Doc is used to construct virtual documents for each entity in the ontologies and Vector Space Model is used to compute similarities between the virtual documents. The virtual document of an entity consists of a collection of words extracted from the entity's name, labels and comments as well as the ones from all neighbors of this entity. The Graph Matching algorithm GMO is used to generate RDF bipartite graphs to represent ontologies, and measuring their structural similarity by a new similarity propagation measurement (Rehman and Saba, 2012a,b,c).

In this paper, Domain Problem Ontology can be constructed based on the domain document using the Entity, Operation and O-RGPS that are taken from the document by using the Stanford POS tagger. Similarly the Requirement Sign Ontology is also constructed based on the requirement document(Cregan A., 2008)(Ke-Qing, 2008). The connecting ontologies are constructed based on the semantic similarity between the domain and requirement models (Heflin ,2000)(Karim S, 2007). In O-RGPS domain modeling method, the first step is to construct domain ontologies: domain entity ontology that describes the entity concepts in a domain and the relationships among these concept, domain operation ontology that describes the operation concepts in a domain and the relationships among these concepts. Domain problem ontology model includes the role model, goal model, process model and service model (Cregan A., 2008). DPO defines the relationship with the domain ontologies (Alipanah, 2012) (Seidenberg J, 2006). On the other hand, DPO can cover the general information of RGPS domain models, so as to relate the domain ontologies with domain models, and provide integrated solution for domain problems. In this way, the domain assets can be modeled and managed semantically in order to improve the reuse efficiency and quality of domain assets. If the semantic similarity between two models is low, it implies that the potential of exchanging significant data between two interoperation parties is also quite weak, which consequently fabricates ineffective collaboration results. The process of semantic matching between ontology models is composed of following steps: first, perform the semantic matching for all the concepts; second, calculate the semantic matching capability between ontology models (Ehrig M, 2005).

Experiment and Results

The High Level Documents are collected from customers of the project and Low Level Documents are described by the project developers. These documents are collected together and parsing is performed and the resulting tokens are obtained. This process is called as word count. Once the word count is obtained as a result of parsing, then from the resulting tokens stop words are removed. This process is done as Stop word removal. Then the ambiguities are identified and classified .The output of the Ambiguity detection module is compared with "ARKref NP coreference system" . The ARKref coreference resolution system is implemented in java and is available in the web. The ARKref resolution system uses the BNC corpora, web corpora and Wordnet to identify the NP coreferences among the NPs in the sentences. The result of the comparison is shown in Table III. The inputs for the evaluation are taken from the requirements dataset collected from RE@UTS website.

Table III Results Of Coreference Resolution

S.NO	NUMBER OF INPUT SENTENCES	ACTUAL NP COREFERENCE	NP COREFERENCES DETECTED	
			PROPOSED SYSTEM	ARKREF SYSTEM
1	11	3	3	2
2	10	8	7	7
3	10	6	6	4

For the resulting tokens, after elimination of ambiguities, Porter's algorithm is applied and stemming is performed (Giuseppe Lami, 2005).Finally stemmed tokens are gathered and term frequency $TF_{t,d}$ and inverse document frequency IDF_t are calculated and TF-IDF value for each term are found. Using similarity and TF-IDF values semantic graph is constructed and number of times the meaningful term's occurrences in the document is known as page rank and the terms are sorted and ontologies are extracted. In this paper, Ontological extraction of unambiguous requirements based on semantic ranking and significance is proposed. This is compared against Simple relevance based algorithms. Among these algorithms our method provides better results for extraction of significant ontologies from the documents. Precision and Recall are calculated. Results reveal that the proposed model gives better results when number of terms is greater than 60. Using our method, significant ontologies are extracted with higher precision. The Precision and Recall values are shown in Table IV.

Table IV Precision And Recall Values

NUMBER OF TERMS	SIMPLE SIGNIFICANCE BASED EXTRACTION		SEMANTIC SIGNIFICANCE BASED AMBIGUITY FREE EXTRACTION	
	PRECISION	RECALL	PRECISION	RECALL
0	100	0	100	0
20	86	1.0	92	1.3
40	72	1.3	85	2.3
60	56	1.4	70	2.6
80	45	2.0	63	2.7
100	30	2.5	65	3.5
120	28	3.0	58	4.3
140	27	3.1	58	5.6
160	25	3.3	56	7.0
180	24	4.1	56	7.2
200	20	5	55	8.8

From the table, it is observed that as the number of terms increase, the proposed technique shows better Precision values than the relevance based abstraction. It is also observed that as the number of terms increase, the proposed technique shows better Recall values than the relevance based abstraction.

Conclusion

As far as Software Engineering is considered, the SRS produced should be complete in order to deliver a quality product. The technique proposed here effectively extracts the single and multiword terms significant in the domain. The document created as reference helps the analyst to get knowledge about the domain. Then ambiguity detection and classification is applied to remove the ambiguities and hence the unambiguous semantic ontologies are extracted. Semantic ranking algorithm had been applied and semantic graph depicted the relationship between the ontologies.

REFERENCES

[1] Daniel M. Berry, Erik Kamsties, Michael M. Krieger. From Contract Drafting to Software Specification. Linguistic Sources of Ambiguity, University of Waterloo, Ontario, Canada, 2003.

[2] Eduard Hovy. Automated Text Summarization. R. Mitkov (ed). The Oxford Handbook of Computational Linguistics. 2005.

[3] Hakim Sultanov, Jane Huffman Hayes, Wei-Keat kong. Application of swarm techniques to Requirement tracing. Requirements Engineering Journal. Springerlink. May 2011.

[4] Saba, T. and Rehman, A. (2012). Effects of Artificially Intelligent Tools on Pattern Recognition, International Journal of Machine Learning and Cybernetics, vol. 4(2), pp. 155-162.

[5] Ananiadou S.A methodology for automatic term recognition. Proceedings of the 15th conference on computational linguistics. Association for Computational Linguistics. Morristown, NJ, USA, 1994.

[6] Brigitte Orliac, Mike Dillinger. Collocation Extraction for Machine Translation. University of Montreal, Montreal, Canada, 2003.

[7] Saba, T. Rehman, A. and Elarbi-Boudihir, M. (2011). "Methods and Strategies on off-line Cursive Touched Characters Segmentation: A Directional Review". Artificial Intelligence Review, DOI 10.1007/s10462-011-9271-5.pp:45-54.

[8] Diana Maynard, Sophia Ananiadon .Identifying Contextual Information for Multi-Word Term

Extraction. Manchester Metropolitan University, Manchester.,1999.
[9] Ricardo Gacitua, Pete Sawyer, Vincenzo Gervasi. Relevance-based abstraction identification: technique and evaluation. Requirements Engineering Journal. Springer-Verlag London Limited. 2011.
[10]George Tsatsaronis, Irakilis Varlamis, Kjetil Nervag. Semantic Rank: Ranking Keywords and Sentences Using Semantic Graphs. Proceedings of the 23rd international conference on computational linguistics (Coling 2010). 2010.pp 1074-1082.
[11]Rehman, A. and Saba, T. (2011). "Performance Analysis of Segmentation Approach for Cursive Handwritten Word Recognition on Benchmark Database". Digital Signal Processing, vol. 21(3), pp. 486-490.
[12] Rehman,A. Alqahtani, S. Altameem, A. Saba, T. (2013) Virtual Machine Security Challenges: Case Studies, International Journal of Machine Learning and Cybernatics, DOI 10.1007/s13042-013-0166-4.
[13]Chinatsu Aone , Scott William Bennett.Applying machine learning to anaphora resolution. Connectionist, statistical and symbolic approaches to learning for natural language processing.1996. 302–314.
[14]Hui Yang, Anne de Roeck, Vincenzo Gervasi, Alistair Willis, Bashar Nuseibeh.Analyzing anaphoric ambiguity in natural language requirement. Requirements Engineering Journal.2011. Volume 16 163-189.
[15]Porter M.An algorithm for suffix stripping. Program. 1980.vol. 14, pp. 130-137.
[16]Vincent Ng, Claire Cardie. Improving machine learning approaches to co reference resolution. Proc. the 40th annual meeting of the Association for Computational Linguistics 104–111. 2002.
[17]Wee Meng Soon, Hwee Tou Ng, Daniel Chung Yong Lim.A machine learning approach to co reference resolution of noun phrases, Computational Linguistics. Special issue on computational anaphora resolution archive.2001. Volume 27, 521-544.
[18]Ido Dagan, Alan Itai. Automatic processing of large corpora for the resolution of anaphora references. Proc. the 13th international conference on Computational Linguistics 1–3, 1990.
[19]Saba, T. and Rehman,A. 2012a, Machine Learning and Script Recognition, Lambert Academic Publisher, ISBN-10: 3659111708, pp: 210-221.
[20]Susan E.Brennan, Marilyn W.Friedman, Carl J.Pollard.A centering approach to pronouns. Proc. the 25th annual meeting of the Association for Computational Linguistics (ACL) 155–162. 1987.
[21]Yue, Kun, Liu, Wei-Yi, Zhou, Li-Ping.Automatic keyword extraction from documents based on multiple content-based measures. International Journal of Computer Systems Science & Engineering. 2011.Vol. 26, no. 2, pp. 133-145.
[22]Rehman, A. and Saba, T. (2012a). "Off-line Cursive Script Recognition: Current Advances, Comparisons and Remaining Problems". Artificial Intelligence Review, vol. 37(4), pp:261-268. DOI: 10.1007/s10462-011-9229-7.
[23] Rehman, A. and Saba, T. (2012b) "Neural Network for Document Image Preprocessing" Artificial Intelligence Review, DOI: 10.1007/s10462-012-9337-z.
[24]Diaz-Aviles E, Nejdl W, Schmidt-Thieme L. Swarming to rank for information retrieval. Proc. the 11th annual conference on genetic and evolutionary computation.2009.pp 9–16.
[25] Cregan A.W3C semantic Web ontology languages: OWL and RDF tutorial. ISO/IEC JTC1 SC32 11th Open Forum on Metadata Registries, Tutorial, Sydney, Australia, May 19-22. 2008.
[26]Ke-Qing He, Jian Wang, Peng Liang.Semantic Interoperability Aggregation in Service Requirement Refinement. Journal of computer science and technology. 2008.
[27]Heflin J, Hendler J.Semantic interoperability on the Web.Proc.Extreme Markup Languages 2000.Alexandria, USA, Aug.15-18, 2000.
[28]Karim S, Latif K, Tjoa A M.Providing universal accessibility using connecting ontologies: A holistic approach. Proc. 4th Int. Conf.Universal Access in Human Computer Interaction, Beijing, China, Jul. 22-27. 2007.
[29]Neda Alipanah, Latifur Khan and Bjavani Thurisingham. Optimized ontology-driven query expansion using map-reduce framework to facilitate federated queries. International Journal of Computer Systems Science & Engineering . Volume 27. 2012.
[30]Seidenberg J, Rector A.Web ontology segmentation: Analysis, classification and use. Proc. the 15th International Conference on World Wide Web, Southampton, UK, May 11-14. 2006.
[31]Ehrig M, Euzenat J.Generalizing precision and recall for evaluating ontology matching. Proc. the 4th International Semantic Web Conference (ISWC), Poster Session, Galway, Ireland. Nov. 6-10. 2005.
[32]Rehman, A. and Saba, T. (2012c). "Evaluation of Artificial Intelligent Techniques to Secure Information in Enterprises". Artificial Intelligence Review, DOI. 10.1007/s10462-012-9372-9.

3/28/2013