International Journal of Scientific & Engineering Research, Volume 4, Issue 12, December-2013
ISSN 2229-5518

Automatic Evaluation Of Computer Science thesis Using Domain Ontology

Mohammed Muzaffar Hussain, Dr. S.K. Srivatsa

Abstract— Because of advanced in word process technologies, spell checking and grammar checks have become instant. Nonetheless, checking abstract errors from any given text has received little attention from similar researches. An example for abstract error is falsehood. The detection of abstract errors on any plain text is a challenging task for computers. Two of the major challenges are the unstructured nature of text documents and the lack of representation of common sense and domain specific knowledge in machine understandable format. Researches show that thousands of people are submitted thesis per annum in which many conceptual error due to technical errors. Technical errors are mainly caused by the omission of facts about the domain or subject. This paper analyses sample computer Science Thesis notes to find patterns that could be explored towards using information encoded in plain computer notes for the detection of conceptual errors. The paper proposes an ontology based architecture for a conceptual error detection system on computer notes.

Index Terms— Thesis, Computer Science Thesis, Conceptual Error Introduction, Domain, Domain Ontology, abstract error, Ontology

1 INTRODUCTION

Word processors are used in enterprise, homes and learning institutions [2]. One of the plus of modern word processing software is the ability to point out spelling and grammatical errors. New researches on spelling and grammar checkers have focused on detecting erroneously spelled words even though they exist in the dictionary [1]. A common example to what is caught by a recent spell checker is homonym (a word spelled correctly but at a wrong context). Example: I will meat you tomorrow. When most trivial spelling and grammatical errors are avoided by the usage of common word processing tools [3], verifying the conceptual correctness of the written material is totally left to the author. Checking and finding efforts so far have not gone beyond the word and sentence level [1]. Fit spelled and grammatically correct statements may have conceptual errors. First, various content in the same text could contradict with each other. Second, any statement could contradict with world known fact. Currently, little work has been done to automatically detect such abstract errors. Recent researches in the field of Information Extraction show that different approaches can be used to extract the concept out of plain text [4][5]. But the focus of these researches was only with the information extraction and not in validation of the represented concepts. Our hypothesis is that domain ontology can be used to validate concepts. Ontology is defined as the representation of different domain knowledge in machine process-able format [6]. Basically, Abstract errors can be defined as contradictions that occur between concepts that are represented in a given textual document. Contradictions occur whenever information that is communicated in one or more sentences is matched. Incompatibilities are manifested in many ways [7]. Contradictions arise from relatively obvious features such as antonymy, negation, or numeric mismatches. They also arise from complex differences in the structure of assertions, discrepancies based on world-knowledge [8]. Based on the origin and strength of knowledge necessary to validate any given sentence, contradictions could be classified in to two categories.

2 OBJECTIVES

The main objective of this Paper is to examine conceptual errors unmistakable in computer thesis notes and unlock information encoded in plain text computer thesis documents to concept validation. Recognizing different patterns eminent in computer thesis notes, the general objective of this research is to explore those patters to improve the general quality of computer thesis documentations. Hence, the proof of concept prototype constructed to demonstrate this thesis should be able to detect different types of conceptual computer thesis errors from a set of sentences extracted from sample Computer Thesis notes. This research should also focus on possible ways for the extraction of formal and machine process-able knowledge from the unstructured plain computer thesis notes. This should allow for reasoning towards detection of conceptual errors.

3 Scope

This research will mainly focus on finding approaches to conceptually validate plain text computer thesis notes. It will not deal with spelling or grammatical errors as those are dealt with other researches [1]. The scope of this research is limited to computer thesis notes (documents) written in English. Even though conceptual errors exist in paragraphs and even in the bigger document, this research will focus on the basic reasoning task and is limited to validation of concepts at the sentence level.

4 Methodology

To achieve the main goals of this paper, a number of methodologies were applied. To gain a deeper understanding of the problem as well as to explore the possibilities, literature review was performed on two related areas. The major activities performed at this phase of the research are:

Review of literature on avoidable computer thesis errors, contraindications, their causes and classification.

Review of literature on the logical definition of contradictions, the behavior and work performed on the detection of contradictions from natural language processing task point of view.

On the second phase of the research work, analysis of sample computer thesis notes was performed with the help of domain experts. At this phase, 50 sample narrative computer Thesis discharge summaries were analyzed to study the pattern of appearance of important computer thesis concepts in computer notes.

A model and architecture to the validation of most significant conceptual errors in computer thesis was proposed after the analysis of the problem and the structure of computer thesis notes. A proof of concept prototype was built following the architecture proposed on the previous phase. The outcome of the research is then evaluated on the proof of concept prototype. The evaluation was mainly performed to check the robustness of the system.

5 Literature Review

This section starts with a brief summary of literatures on preventable Computer Thesis errors. Related research areas and approaches to contradiction detection are also reviewed.

6 Computer Science Thesis Errors

In a year many computer thesis are made. In that many thesis are giving wrong information which leads to bad research.

7 Contradictions

Little work has been done towards detection of contradiction. However, [8] observed that contradiction occurs when two sentences are extremely unlikely to be true simultaneously. For two sentences to be contradictory, they need to refer to the same event. However, determining if two sentences are co-referent is probabilistic rather than certain. This problem was also identified by [9] as reference resolution problem. Two categories of contradictions have been identified by [8]. The first category includes contradictions that occur because of the usage of antonyms, negation and numeric features. This category of contradiction is relatively easy to detect. The second category of contradictions contains contradictions that need world knowledge or commonsense knowledge to detect. Textual Entailment is formally defined as a relationship between a coherent text T, and the hypothesis H. T is said to entail H (T → H) if the meaning of H can be inferred from the meaning of T [2]. Textual entailment recognition is therefore the process of determining if a given natural language text is inferred from semantic of another one. [9] Recognized detection of entailment and contradiction relation between texts as a minimal matrix for the evaluation of text understanding. After this observation, Recognition of Textual Entailment (RTE) came as a research area that focuses on the sole task of recognition of entailment between a given text and a Hypothesis.

8 Logical Inference

[11] Presented a RTE system that works by using logical inference. First, the authors used a system called BLUE (Boeing Language Understanding Engine) to perform a full semantic interpretation of both sentences. Then, they used knowledge obtained from WordNet and DIRT paraphrase database to infer a relationship. WordNet is one of the most used lexical resources in NLP. It organizes words in semantic networks: the nodes, synsets, represent senses, and contain a number of single or multi-word terms which have the same or very similar meaning; the edges represent different types of semantic relations, such as hyponym-hypernym,

9 Ontology Alignment for RTE

[12] with their submission to the RTE 4 challenge presented a system that works by aligning ontology"s acquired from the given text (T) and hypothesis (H). To achieve this, their system performed three separate processes.

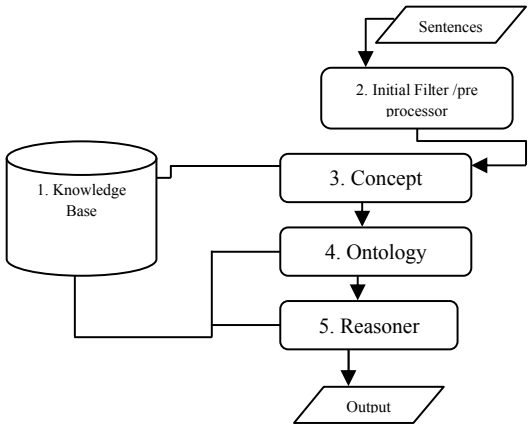

Fig 1 Architecture of conceptual validation system

10 Ontology Acquisition

On this phase, The system finds a formal representation of the given text (T) and the Hypothesis (H). This formal representation is based on a description

logic type of ontology. To generate the formal representation of the two texts, the system first performs a syntactic analysis with a parser called Minipar. Minipar is used to generate the dependency relations. These dependency relations are transformed to a semi-Semantic structure using a set of transformation rules

11 Ontology Alignment

After the two ontology‟s are generated on the ontology acquisition phase, the RTE System performs a two-step operation to align them to create an aligned ontology Ontology-A. First, classes are aligned to create equivalent classes. On the second step, the properties are aligned to create equivalent properties.

12 Textual Entailment

Data collected on the ontology alignment phase is used to decide if there is a textual entailment relation or not. To achieve this, the system integrates WEKA, an open source machine learning package, to make the decision if there is an entailment relation or not. WEKA is trained with RTE3 test set. The authors selected three machine learning algorithms for their entry on the RTE4 completion. WEKA B40 decision tree classifier was however the one algorithm that resulted in the best performance of 68% on their 2 way RTE4 challenge submission. Even though all the above approaches have been applied on the RTE task, [13] argues that there is a tradeoff between informatively and robustness. **Informatively** is the ability of a system to take into account all available relevant information. **Robustness** is the ability of a system to proceed on reasonable assumptions, where relevant information is missing.

13 Architecture

This section discusses the architecture of the proposed concept validation system and shown in fig 1. The proposed architecture attempts to explore the pattern observed on the analysis of the computer thesis documents. The pre-processor, the knowledgebase, the ontology extractor and the reasoner are the main components in the architecture of the concept validation system. The

Input to the concept validation system is a sequence of sentences. These sentences are initially passed to the pre-processor. The pre-processor is responsible for filtering out sentences deemed un-important in the concept validation process. The knowledge base is a store for background knowledge as well as restrictions that are applied in the concept validation. The background knowledge in the knowledge base component of the concept validation system is a standard representation of computer concepts. This standard representation along with the standard classifications that apply on the represented concepts will be used in different stages of the concept validation process. After the pre-processing, the sentences are passed to the concept mapping component. The concept mapping component is responsible to identifying computer concepts from the sentences and mapping them to the standard representation in the knowledge base. The output of the concept mapping component is passed to the ontology extractor. The ontology extractor is a component in the concept validation system that constructs ontology of the concepts represented in the computer note. The very reason for extracting the ontology from the text is to reason on a structured representation of the information in the unstructured plain text. As can be seen in figure 1 the output of the ontology extractor is then passed to the reasoner.

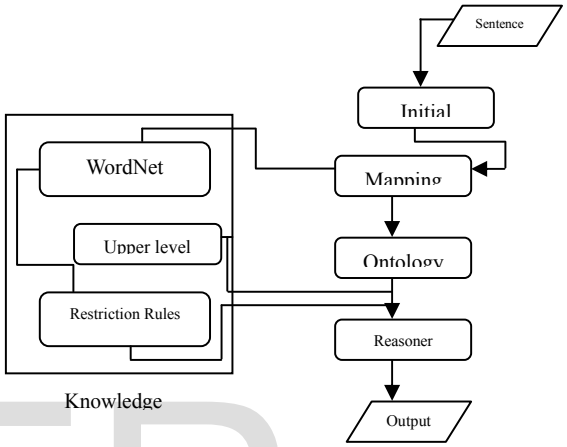

Fig 2 Implementation of Architecture

14 Implementation

In accordance with the architecture of the proposed system, Figure 6.1 shows what components were used to perform the functionalities described in the fig 2**.**

14.1 The Knowledge Base

The knowledge base component of the system is composed of three separate and interoperable components. The first one is the WordNet. **WordNet** is a lexical database for the English language [1]. It groups English words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between these synonym sets. The purpose is twofold: to produce a combination of dictionary and thesaurus that is more intuitively

usable, and to support automatic text analysis and artificial intelligence applications The hypernym/hyponym relationships among the noun synsets can be interpreted as specialization relations between conceptual categories. In other words, WordNet can be interpreted and used as a lexical ontology in the computer science sense. However, such an ontology should normally be corrected before being used since it

contains hundreds of basic semantic inconsistencies such as (I) the existence of common specializations for exclusive categories and (ii) redundancies in the specialization hierarchy. Furthermore, transforming WordNet into a lexical ontology usable for knowledge representation should normally also involve (i) distinguishing the specialization relations into *subtypeOf* and *instanceOf* relations, and (ii) associating intuitive unique identifiers to each category. Although such corrections and transformations have been performed and documented as part of the integration of WordNet 1.7 into the cooperatively updatable knowledge base of WebKB-2, most projects claiming to re-use WordNet for knowledge-based applications (typically, knowledge-oriented information retrieval) simply re-use it directly. WordNet has also been converted to a formal specification, by means of a hybrid bottom-up top-down methodology to automatically extract association relations from WordNet, and interpret these associations in terms of a set of conceptual relations, formally defined in the DOLCE foundational ontology. [4]

14.2 Initial Filter

The input sentences pass through an initial filtering step that discards out non informative sentences. Even though they are rare, these kinds of sentences could create a noise on the final output of the system. Good example for non informative sentences is question. This component is also responsible to identify and discarding sentences that are not written about the current Thesis. From the analysis of the sample documents we have learnt that some sentences in the Computer thesis note describe or discuss the computer thesis condition of the previous used thesis. In extreme cases, we have noticed that some sentences are written about other thesis. Hence this component of the concept validation system is responsible to detect and discard such sentences. In real world settings, we recognize that reference resolution shall involve further steps. These steps include construction of an indexed database of all subjects. Such a database will be used to recognize co-referent sentences from within large documents. But, as stated in the scope of this document, this work only concentrates on the other components of concept validation system. Under such a setting, the distance and location of sentences can be taken into consideration to facilitate the decision making. Sentences in the same paragraph are more likely to discuss about the same idea than sentences in separate or far apart paragraphs. Assuming that non co-referent sentences cannot contradict, the system avoids processing them further. The implementation of the prototype automatically assumes that sentences containing words that represent any previous thesis.

14.3 Mapping

The unstructured representation of knowledge in the sentences is difficult for the reasoning task we want to perform. To address this problem we need to find a tool that extracts only important concepts from the text into a standard conceptual representation. The most commonly used tool in the computer domain to this end is mapping. After the initial filter is performed, the sentences that are deemed important are passed to Mapping. Mapping breaks down the sentences into phrases. For each phrase in the sentence, it returns the mapped concept from WordNet ranked by the mapping strength. The output of Mapping is an XML document representing each concept that has found a mapping in the WordNet. The XML document also contains the semantic group of the mapped concept. Hence from the semantic group of the mapped concept.

14.4 Ontology Extraction

The XML output of Mapping is consumed by the ontology extractor. At this stage, ontology of the statements is acquired for further analysis. For the prototype, we have selected two types of important Information for the validation of concepts. Some of these extractions not necessarily build the information from scratch. Some enhance previously extracted and constructed knowledge about the current Thesis. Naturally, this data could be taken from different parts of different documents.

1. Current Thesis Information: Known All New concepts used,

2. Previous thesis information: including thesis name, concepts of the thesis,

Taking the assumption that the subjected computer science thesis note is written all about a Thesis, the ontology extraction tool initialize a "current Thesis" object that is of type thesis. After creating an ontology instance of "current Thesis" in the upper level ontology, the ontology extractor component goes on to parse the xml output of MetaMap. Using XPath expressions, the ontology extraction tool selects all concepts that fall under the computer semantic type."//Phrase/Mappings/Mapping/Candidates/Candidate[STs/ST='dsyn'/UMLSCUI/text()"Execution of the above XPATH expression returns all wordnet concept IDs that are of semantic type thesis. NegEx comes with the regular expression patterns of pre-negations and post-negations. Pre-Negations are negations that come before the negated word

14.5 Reasoner

The last component in the concept validation architecture is the reasoner. The reasoner is mainly responsible for checking if the restriction rules in the restriction rule base are not satisfied. Hence, this component contains two sub-components. The rule factory: this is a component of the system that translates the rule from the representation in the rule database to an ontology reasoning format. First this component selects the entries that involve the disease and clinical drugs that have instances in the extracted ontology. This functionality makes it easier for the system to execute and

check only the important rules to be checked making the system efficient in light of a big rule database. The rule factory is one of the extendable features of the concept validation system. Extending the rule database to include new types of rules would require that the interpretation of the new restriction rule type being implemented in this rule factory. After selecting only the important rules from the database, the rule factory will translate the rules into SPARQL query. It creates a "check list" or rules that might have been violated. To this end, the rule factory selects out only rules that involve instances of concepts in the extracted ontology.

14.6 User Interface

A simple graphical user interface was constructed to simplify the usage of the concept validation system. Figure 3 is a screen shot of the graphical user interface for the concept validation system. The input to the concept validation system is a free text paragraph of computer thesis narration.

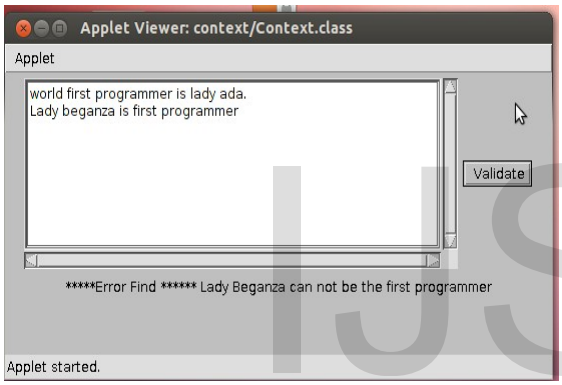

Fig 3 GUI Of Context validation

15 Conclusion

Previous researches have shown that computer thesis errors result in thousands of erroneous in the United States alone. Most of these errors are deemed preventable as they occur because of omission of some kind of information. Some of the efforts towards avoiding these errors from the computer usage perspective are usage of computer data order entry (CDOE) .The systems however depend on the entry of structured data. The very fact that CDOE depend on structured data renders knowledge represented in plain text previous

computer thesis un-usable towards the validation of contraindications and contradictions. This paper has explored usage of available natural language processing tools and approaches combined with domain specific knowledge towards unlocking information contained in the computer thesis notes. An approach to extract knowledge from the plain text computer thesis notes was developed. Together with the applied background knowledge, a restriction rule data store was also developed. Later, a reasoning component was developed to use the background as well as domain specific restriction rules towards detecting conceptual error.

16 Future works

The results of this paper have demonstrated that a concept validation system could unlock information encoded in plain text documents towards detection of conceptual errors. However, this work could mainly benefit from integration of different areas of researches. This section lists a brief list of areas of improvements for this paper work.

1. The concept validation system would be improved by Incorporation of time frame detection approach and algorithm.

2. Expanding and refining the concept extraction component to include other Departments such as Electrical, communication to the validation process.

3. Adding additional rule sets to accommodate the additional variable types.

17 References

[1] Golding, Andrew, and Dan Roth, A Winnow-Based Approach to Context-Sensitive Spelling Correction, Machine Learning 34.1 (1999): 107-130.

[2] Wiki 2009, http://en.wikipedia.org/wiki/Word_processor, last accessed on May 3, 2009.

[3] Haigh, T., Remembering the Office of the Future: The Origins of Word Processing and Office Automation. Annals of the History of Computing, IEEE 28.4 (2006): 6-31.

[4] Bennett, N. A., He, Q., Chang, C., & Schatz, B. R., Concept extraction in the interspace prototype, Urbana-Champaign, IL: CANIS - Community Systems Laboratory, University of Illinois at Urbana-Champaign, Champaign, IL.

[5] Burget, R., Layout Based Information Extraction from HTML Documents." DocumentAnalysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on 2(2007): 624-628.

[6] Wiki1 2009, http://www-ksl.stanford.edu/kst/what-is-an-ontology.html, last accessed on June 7,2009

[7] Sanda M. Harabagiu, Andrew Hickl, and V. Finley Lacatusu. Negation, Contrast and Contradiction in Text Processing, 2006, in proceedings of AAAI-06.

[8] De Marneffe, Marie-Catherine, Anna N. Rafferty & Christopher D. Manning, Finding contradictions in text. In Proceedings of the Association for

Computational Linguistics 2008, 1039–1047. Columbus, OH: Association for Computational Linguistics.

[9] Dick Crouch, Cleo Condoravdi, Valeria de Paiva, Reinhard Stolle, and Daniel G.Bobrow, Entailment, intensionality and text understanding. In Graeme Hirst and Sergei Nirenburg, editors, HLT-NAACL 2003 Workshop: Text Meaning, Edmonton,

[10] Wiki2 2009, http://ai-nlp.info.uniroma2.it/te/, last accessed on June 2, 2009

[11] Johan Bos , Katja Markert, Recognizing textualentailmentwithlogicalinference,Proceedings of the conference on Human LanguageTechnology and Empirical Methods in Natural Language Processing, p.628-635, October 06-08, 2005,

[12] R. Siblini and L. Kosseim, Using Ontology Alignment for the TAC RTE Challenge, Notebook Papers and Results, Text Analysis Conference (TAC-2008),2008,301-309

[13] Bergmair, R, Monte Carlo Semantics: MCPIET at RTE4, In Text Analysis Conference(TAC 2008) Workshop - RTE-4 Track. National Institute of Standards and Technology (NIST). November 17-19, 2008. Gaithersburg, Maryland, USA.