Testing the Accuracy of Text Deconstruction Using PTree Tool

Wan Malini Wan Isa (Corresponding Author)
Department of Multimedia, Faculty of Informatics
Universiti Sultan Zainal Abidin, Gong Badak Campus
21300 Kuala Terengganu, Terengganu, Malaysia
Phone: 609-6664466 E-mail: wanmalini@udm.edu.my

Jamaliah Abdul Hamid
Department of Foundations of Education, Faculty of Education Studies
Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
E-mail: aliah@putra.upm.edu.my

Hamidah Ibrahim
Department of Computer Science
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
E-mail: hamidah@fsktm.upm.edu.my

Mohd Hasan Selamat, Rusli Abdullah
Department of Information System
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
E-mail: hasan, rusli {@fsktm.upm.edu.my}

Nurul Amelina Nasharuddin
Department of Multimedia
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
E-mail: amelina@fsktm.upm.edu.my

Abstract

Our research project is currently to develop an Automatic Concept Relation Extraction (ACRE) System which automatically extracts concepts and their relationships across texts in all domains of knowledge. Concept Relational Tree (CRT) is one of the text analyzer applications used in the ACRE System to automatically extract concepts and their relationships in a document. To check on the correctness of the extraction of concepts and their relationships, the PTree is designed to reconstruct the text by reverse input. In this paper we present the PTree tool to test the accuracy of the automatic tagging and tree structure created by CRT from texts. The PTree tool is implemented from Java Universal Network/ Graph Framework (JUNG) libraries. This tool provides a few functions to allow for flexibility in drawing parse trees for concept relationships. Due to its flexibility and dynamic features, PTree can be further extended for use in the deconstruction of highly complex texts.

Keywords: Parse tree, Java Universal Network/Graph, Interface

1. Introduction

In today's information explosion it has become more and more necessary to automatically enable the extraction of main concepts and relationships between concepts in a dynamic growing knowledge schema from documents

in the database. The automatic extraction of concept, related concepts and their relationship from selected documents will enable user to obtain knowledge ontology as well as to search in more directed manner the relationship of concepts.

The ACRE System is an ongoing development of a system that extracts concepts and their relationships automatically across domains of knowledge. When fully developed, ACRE is expected to be able to automatically extricate conceptual knowledge from texts for the building of interactive and dynamic knowledge ontologies. At the present stage of development, ACRE is able to partially extract concepts automatically and the relationships between these concepts from a collection of documents. ACRE also enables visualization of the network of concept relations (Nasharuddin *et al.,* 2008), and the trace-back of selected concept-relation schema back to its original source in a document. Although still in the early stage of development, ACRE shows potential contribution into research on automatic extraction of concepts and their relationship, in lieu of the more time consuming method of machine learning or rule based algorithm, or the laborious process of expert-dependent input.

1.1 Brief description of the text analyzer components used in ACRE

In the process to extract concepts from text and their relational mapping, the ACRE system uses a few text analyzer components which are Concept Relational Tree (CRT), Connector Based Extraction (CBE), Concept Relational Parser Tree (CRPT), Concept Relational Model (CRM) and Social Competition Model (SCM) (Abdullah *et al.,* 2008, Ungku Chulan, 2007). When ACRE deconstructs text, it performs CRM sentence tagging based on a three classes: concepts, relations and attributes, instead of the usual complex categories of POS (Part of Speech) tags. CRT then plays a crucial role in arranging these concepts, relations and attributes so that the semantic hierarchy is maintained even as texts get more complex. CRT is essentially an integration of Discourse Structure Tree (DST) and the Expression Tree (ET). Essentially, DST analyzes text markers as a basis to organize semantics hierarchy between concepts, but ET improves granularity of sustaining the correct hierarchical relationship between concepts by providing a new framework for semantic organization based on connectors rather than discourse markers (Ungku Chulan *et al.,* 2008).

Connectors such as verb, prepositions and conjunctions are more stable parts of any sentential text. They occupy very specific hierarchical positions in a sentence. This characteristic makes connectors a preferred choice as the point of semantic connectivity in any text. Thus a connector based extraction (CBE) model will inadvertently increase the accuracy of the extraction of conceptual relationships by CRT.

As a result of CRT, a newly improved parse tree which we call the Concept Relational Parser Tree (CRPT) is built that works on the simple structure that each concept is linked to another, whether in the position of agent or object, by a connector (Ungku Chulan *et al.,* 2008).The connector links one level of the tree to the next. Thus the tree grows vertically, with the original agent concept firmly anchored at the foot of the tree.

One of the greatest challenges in sentence deconstruction is how to deal with sentence nesting. To deconstruct texts containing nesting, relational precedence *o(R)* must first be determined from the hypothetical degree of relatedness between concepts between or among the nesting. In this project, as reported in the doctoral dissertation, a Social Competition Model (SCM) was developed (Ungku Chulan, 2007) to further enhance CRT. The basic principle underpinning SCM is that the relationships are formed amongst concepts as a manifestation of semantic competition.

1.2 The PTree as a testing tool

The CRT component however, needs to be tested in order to show the accuracy of ACRE in performing automatic sentence deconstruction to the level of concepts and their relationships. The CRT also has to be evaluated for its accuracy in extracting concepts and their relationships in nested sentences. This paper will focus on developing PTree as a testing tool to re-construct the sentences parsed by CRT and modeled by the CRPT in the ACRE system (See Figure 1).

Methodologically, PTree enables those concepts and relations identified by CRT to be manually re-inserted by the user at various levels of the parent and child nodes in PTree, and when the traversal order command is executed, PTree should then reconstruct the entire sentence. A point to note however: since CRT has already tagged the original sentence, the tree structure actually produced in CRPT is therefore that of a modified sentence structure, although the semantics or meaning has been retained as much as possible. In testing, the sentence produced by the PTree should therefore resemble closely the modified sentence used in CRPT process. Working backwards, this would prove that the CRT has successfully extricated concepts and their relationships from straightforward and nested types of texts. Another point to note: since CRM, CRT, CBE, SCM, CRPT are

intricately integrated during text deconstruction in ACRE, for the purpose of this paper when we describe the testing of CRT process, we mean the integration of all these components.

2. The construction of PTree: Method

Parse tree is commonly generated for sentences in natural languages, as well as during processing of computer languages, such as programming languages. A parse tree is a tree that arranges the words in the sentence according to their part-of-speech tag and production rules (See Figure 2). The production rules determine the hierarchical manner of which tags are related to one another by specifying the formula of tag decomposition.

The PTree combines the concepts of binary and conventional parse tree. Like the conventional binary tree, the PTree has a parent node that has two children in the left and right positions but unlike it, the PTree allows for the entry of string entity instead of just numbers to label the child nodes and the leaves. This labeling feature is important since the leaves of a tree represent concepts extracted from texts, while the relations are depicted as the branch child nodes. The PTree uses the basic C-R-C tree structure used in CRT and in this way, the PTree economizes the sentence tree without affecting the meaning of the sentence (Selamat *et al.,* 2008).

The PTree's function as a testing tool begins after the CRM has custom tagged the parts of speech. PTree inserts the concepts and relations identified in the CRM into its nodes, in the order suggested by the CRPT based on the C-R-C platform. At the present stage the input entry is still manual, but work is being carried out to automatically insert the elements, with the automatic build up of subsequent layers of trees. Once all entries are completed, the traversal order command is executed and the sentence will be re-constructed. Perfect score is achieved when the re-constructed sentence produced by PTree sentence matches the de-constructed sentence of the CRT.

2.1 Testing the accuracy of CRT using the PTree

The test of the accuracy of CRT using the PTree tool was done using 100 sentences selected randomly from material collected by Newsblater (http://newsblaster.cs.columbia.edu/). PTree tests 5 steps of the entire CRT text deconstruction process of ACRE consisting CRM tagging, CRT extraction of concepts and relationships, and CRPT tree production.

Step 1: Sentence tagging. Sentence used for the testing will be first tagged by ACRE using normal part of speech tagging consisting of noun (NN), verb (VB), determiner (DT), and adjective (JJ) and many more.

Original sentence: After a touchback, Scheffler caught a 16-yard pass before hopping off the field with an apparent right foot injury.

>> RESULT after sentence tagging: After/IN a/DT touchback./NN Scheffler/NNP caught/VBD a/DT 16-yard/JJ pass/NN before/IN hopping/VBG off/RP the/DT field/NN with/IN an/DT apparent/JJ right/JJ foot/NN injury./NN

Step 2: Sentence modification by ACRE. The tagged sentence is modified to make it compliant to CRM tagging so that a tree base structure of C-R-C is produced.

>>RESULT after sentence modification: Scheffler caught a 16-yard pass before hopping off the field with an apparent right foot injury after a touchback.

Step 3: Based on CRM tagging, the parts of speech are then hierarchically sequenced in a C-R-C tree constructed by the CRPT component in the ACRE system. The CRPT is an important step to ascertain the hierarchy of semantical relationship of concepts and their relations after text deconstruction is completed.

>>RESULT of sentence modeled with CRM: Scheffler/C caught/R a 16-yard pass/C before hopping off/R the field/C with/R an apparent right foot injury/C after/R a touchback/C.

At this point, the test is taken over by PTree. Steps 4 and 5 describe the test details. In these latter steps, the process is reversed whereby the test sentence is reconstructed.

Step 4: Insert the tags from step 3 to the nodes of the PTree in order to perform sentence reconstruction. Leaf nodes may be added on using the function buttons provided in the tool (See Figure 3).

Step 5: Click on the button "In order Traversal" to now automatically execute sentence reconstruction. The in order traversal function in PTree reads all nested sentences and reconstruct them, based on the content of the nodes.

>>RESULT: Sentence produced is nested: ((((Scheffler caught 16 yard pass) before hopping off field) with apparent right foot injury) after touchback) (See Figure 4).

By manually comparing the nested sentence from sentence reconstruction in Step 5 with the modified sentence produced by ACRE in Step 2, one is able to judge the accuracy of CRT.

2.2 Limitation of PTree

One of the limitations of the PTree tool is that it is presently only able to test the accuracy of deconstructed sentences containing up to 4 nested parts within them. Also, the test is only able to reflect partial semantics of the text based on two schemes which are CRC and RCRC. Test results show that 80% of semantic deconstruction of sentences is correct. Another severe limitation manifested by PTree at this present time is that it cannot correctly parse sentences beginning with prepositions. The result does not tally with the result of the sentence produced by ACRE.

3. Description of user functions in PTree

One of the nuisances in tree parsing is the lengthy time it takes to manually build a tree. PTree tool has several functions in order to allow user draw a tree in a simpler way. It contains menu bar, tool bar, tree viewer and text area. When the user runs this tool, one node will appear at the center of the tree viewer which is called the root node. From a single root node, the user can extend the node to become a tree. To build a tree, user right clicks on the node and a pop up menu will display. The pop up menu facilitates the user to interactively draw the tree as shown in Figure 5.

The functions are "**Add node (L&R)**" to add left or right nodes to an existing node, but not both. When the next level of child nodes have been extended, the left or right node of the previous level now changes its type from concept to relation. For example in Figure 6 (left) node 1 is a concept but when user adds a child node to the node, node 1 changes its type from concept to relation. See Figure 6 (right). This allow for new levels of concepts to be introduced and related to one another in the CRC structure. In this way too, the parent node maintains the initial pivotal relation that encapsulates all other relationships spawned by n-levels of nodes. It is this ability of PTree to maintain the pivotal relationship at the parent node that enables the tree to hold on the essential meaning of a sentence in spite of any number of subsidiary concepts and their relationships are added on.

Another function on pop up menu is "**Delete node**". When the user performs this function on a child node, the node will be deleted. If the user performs this function on a parent node, all the child nodes belong to the parent will be deleted including the parent node.

This PTree tool enables the user to input (insert) the content to the node. User can perform this action by double clicking at the node and a pop up input dialog will be displayed. If the user wants to change the content of the node, the user double clicks on the node and enters the new content into the input dialog box. The content of the node are not limited only to numbers or words. In conventional binary trees, the rigid notation by numbers preserves the order of the tree hierarchy, and difficulty arises when sentences become more complex, with many subsidiary concepts.

The tool bar in this PTree has three formatting buttons. The first button is "**Reset**" button. This button is to reset the node to the earlier position, clear all the nodes and their contents and draw a new root node. The next button is "**In-Order Traversal**" button. When user clicks this button, the tool will traverse the tree from left to right node to compute or read all the nodes of the left subtree, the root and lastly the right subtree. As a result of the traversal, the reconstructed sentence will be displayed in the text area.

This tool also has a menu bar which contains "**Save as Image**", "**Print**" and "**Exit**" function.

4. Conclusion

This PTree tool was developed to test the accuracy of CRT in the ACRE system in extracting concepts and relations from sentences. By comparing the reconstructed sentence produced from the PTree to the sentence parsed by CRT, the accuracy of the CRT can be determined. Tests show up to 80% accuracy or match for sentence containing up to 4 nested parts. In conclusion, the PTree has proven itself a useful test to test on the accuracy of CRT in the ACRE system.

References

Abdullah, R., Abdul Hamid, J., Selamat, M.H., Ibrahim, H., Ungku Chulan, U.A.I. (2008). Semantics Representation in a Sentence with Concept Relational Model (CRM). In Proceeding of Knowledge Management International Conference 2008, Langkawi, Malaysia, 10-12 June 2008, Vol. 1, 112-116.

Barthelemy, F., Boullier, P., Deschamp, P., Kaouane, L., Khajour, A., & Clergerie, E.V. (2001). Tool and resources for Tree Adjoining Grammars. In Proceedings of the ACL 2001 Workshop on Human Language Technology and Knowledge Management, 15, 63-70.

Black, P. E. (2007). Binary tree. [Online] Available: http://www.nist.gov/dads/HTML/binarytree.html (January 2008)

Cohen, R. F., Battista, G. D., Tamassia, R., Tollis, I. G. & Bertolazzi, P. (1992). A framework for Dynamic Graph Drawing. In Annual Symposium on Computational Geometry, Proceedings of the eighth annual symposium on Computational geometry.

Gross, J., & Yellen, J. (1999). *Graph Theory and Its Applications.* Boca Raton, Florida: CRC Press.

Hanrahan, P. (2001). To Draw a Tree. In Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'01) (p. 3-3).

Kennedy, J. (n.d), Parse Trees. [Online] Available: http://homepage.smc.edu/kennedy_john/PARSETREES.PDF. (January 2008)

Mitchell, S. (2008). *Binary Trees and BSTs.* [Online] Available: http://msdn2.microsoft.com/en-us/library/ms379572(VS.80).aspx (January 2008)

Moen, S. (1990). Drawing Dynamic Trees. In *IEEE Software*, *7*, 21-28.

Nasharuddin, N.A, Abdul Hamid, J. Ibrahim, H., Selamat, H., Abdullah, R., & Wan Isa, W.M. (2008). Visualizer for Concept Relations in an Automatic Meaning Extraction System. In *The Journal of Information and Knowledge Management Systems*, Vol. 38, No. 2, 232-240.

O'Madadhain, J., Fisher, D., Smyth, P., White, S., & Boey, Y.B. (n.d). (2008). Analysis and visualization of network data using JUNG. In *Journal of Statistical Software*. [Online] Available: http://jung.sourceforge.net/doc/JUNG_journal.pdf. (January 2008)

Santorini, B. (1990). Part of Speech Tagging Guidelines for the Penn Treebank Project. [Online] Available: ftp://ftp.cis.upenn.edu/pub/treebank/doc/**tag**guide.ps.gz (February 2008)

Selamat, H., Wan Isa, W.M., Abdul Hamid, J., Ibrahim, H., Abdullah, R., & Nasharuddin, N.A. (2008). PTree: A tool to draw tree for Concept Relation Tree (CRT). In Proceeding of Knowledge Management International Conference 2008, Langkawi, Malaysia, 10-12 June 2008, Vol. 1, 117-121.

Ungku Chulan, U.A.I. (2007). Connector-based Extraction with Concept Relational Parser for Extracting Semantic Relation from Text. PhD Thesis (Unpublished), Universiti Putra Malaysia.

Ungku Chulan, U.A.I., Sulaiman, M.N., Mahmod, R., Selamat, H., & Abdul Hamid, J. (2008). Organizing the semantic of text with the Concept Relation Tree. In *International Journal of Computer Science and Network Security*, Vol 8, No. 9, 236-249.

Zanden, B.V., & Beeler, M. (n.d). (2008). A Tool for Sketching and Manipulating Binary Heaps. [Online] Available: http://www.cs.utk.edu/~bvz/HeapAnimation.pdf (January 2008)

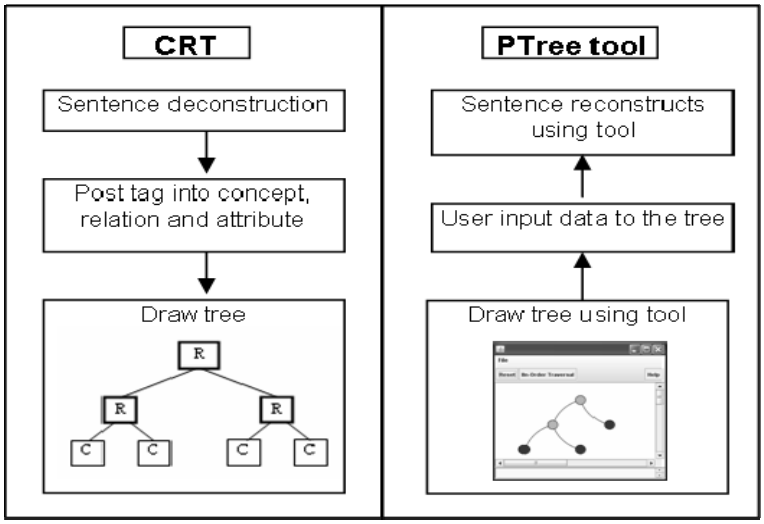

Figure 1. Sentence deconstruction and reconstruction

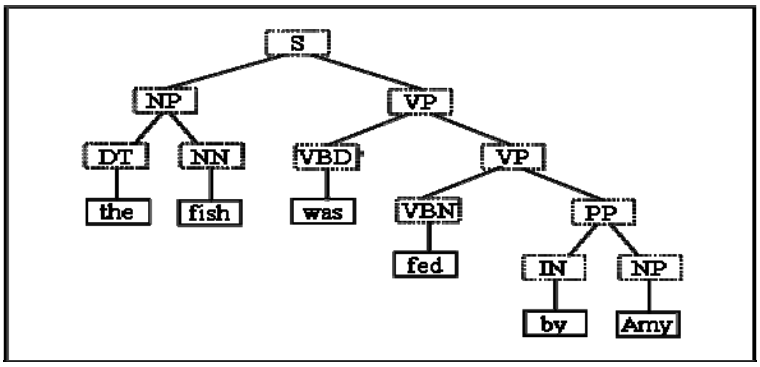

Figure 2. Parse Tree

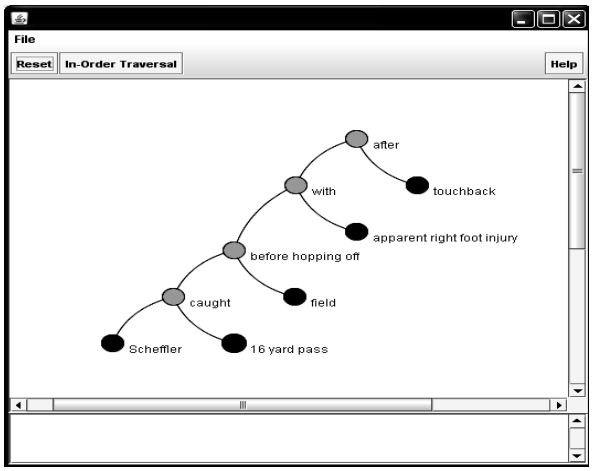

Figure 3. Insertion of tags using PTree tool

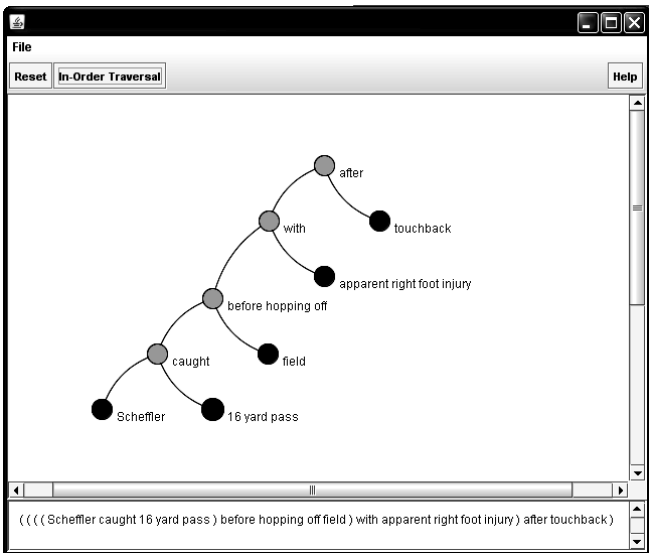

Figure 4. Sentence reconstructed by PTree.

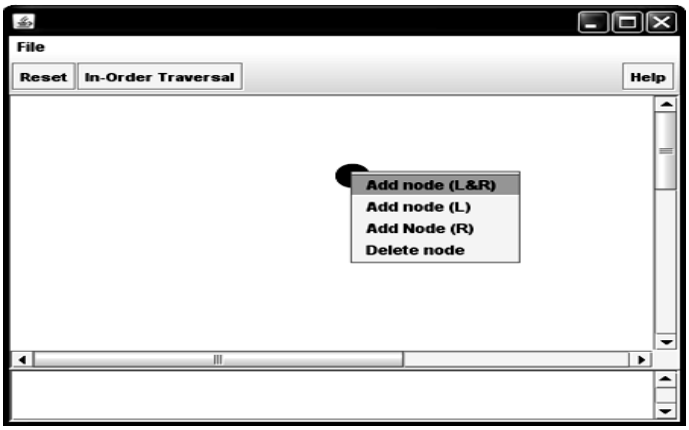

Figure 5. Pop up menu function.

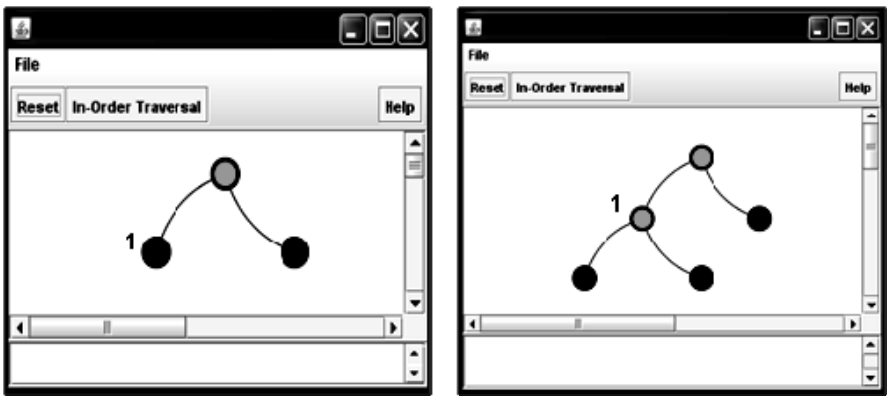

Figure 6. Type node change from concept to relation