

# Visualizing the Meaning of Texts

Wai K. Yeap, Paul Reedy, Kyongho Min and Hilda Ho

*Natural Language Processing Group, Institute for Information Technology Research,  
Auckland University of Technology, Auckland, New Zealand  
{wai.yeap, preedy, kmin, hho}@aut.ac.nz*

## Abstract

We implemented *SmartINFO*, an experimental system for the visualization of the meaning of texts. *SmartINFO* consists of 4 modules: a universal grammar engine (UGE), an anaphora engine, a concept engine and a visualization engine. We discuss two methods of visualizing meanings of text. One approach is a word-centered approach and the other, a clausal-centered approach.

*Keywords*--- text visualization, natural language

## 1. Introduction

How do we implement an information visualization system to visualize the meanings of texts? This is a difficult problem due to the fact that the meaning of a text, unlike the sentences that represent it, is not simply read off the page. It is a complex process to extract meanings. Sentences must be parsed, anaphors resolved and discourse structures computed. Moreover, no adequate solution has been found for each of these steps when dealing with real-world text. Thus, the problem of visualizing the meanings of texts poses a major information visualization problem.

Document visualization systems typically involve the visualization of themes and topics from: large documents [1], document sets [2], or text streams [3]. Such visualizations show frequently occurring themes and their relatedness to each other within a document, or the relatedness between documents within a set. While another system [4] visualizes the entire text of a document. However, with all of these systems, the user can not obtain the intended meaning of the text from the display alone. The systems act more as an overview or a guide. To obtain the actual meaning, a user is still required to read the original text.

We propose to extend this domain by developing a system to visualize the meaning of texts. So that a user will be able to recover the intended meaning of a text

entirely from the interactive visualization system. The purpose of this system is to allow a user to locate and acquire information, quickly and clearly, from the visualization display, without needing to read the underlying text document.

We implemented an experimental system called *SmartINFO* which is designed to investigate this problem. *SmartINFO* is implemented in LISP. Fig. 1 shows the system architecture. It consists of 4 important modules, namely a universal grammar engine (UGE), an anaphora engine, a concept engine and a visualization engine.

The first two modules extract useful information from the input text and the last two modules prepare and visualize the information extracted. This paper focuses on the latter problem and suggests two different methods of creating a concept network to be visualized. The first method is referred to as a word-centered visualization method and the other, a clause-centered visualization method. Sections 3 and 4 discuss these two methods respectively. Section 2 provides some background on the earlier modules. Section 5 concludes the paper with a discussion of results and future work.

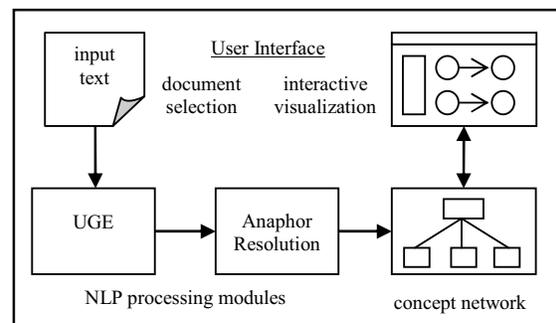


Figure 1: System architecture of *SmartINFO*

## 2. Background

In order to tackle the problem of parsing real-world text, we experimented with a new parser, the UGE.

The UGE parser, developed by Yeap [5,6] is based on an hypothesis of how children acquire their first language. It utilizes the left/right attachment of words as a framework for the processing of language. An example output produced by the UGE is shown below:

(UGE '(then they made the grain into flour by grinding it in a hand-mill))

```
[made* (:actor (they* (:manner (then*))))
(:what (grain* (:modifier (the*))) (:into* (flour*)))
(:by* (grinding* (:what (it*
(:in* (hand-mill* (:modifier (a*)))))))]
```

Note that in most cases, the UGE will generate multiple outputs for a sentence. Using some rules on the role of syntactic information, the UGE will select the best possible interpretation to be passed onto the next module, the anaphor resolution module.

An anaphor is a word that refers to an entity that has been introduced previously. Consider the sentence:

(S1) Keelin likes milk and he drinks it often.

The word “he” is a pronominal anaphor that refers to the word “Keelin”. We have implemented a knowledge-poor anaphora resolution algorithm which is able to find the antecedent for both inter- and intra-sentential anaphora expressions of third person pronouns (e.g. he, she, it, they). The algorithm and its performance compared with others is reported in [7].

Next, the concept engine establishes semantic relationships between all noun terms found in the text. Our first attempt is to capture the surface meaning of each sentence. By displaying these relationships to the users, it is possible to convey some meanings of the text to them. We now discuss two methods to do so.

### 3. Word-centered visualization

Our first approach is centered on making explicit the direct syntactic relations between words. Examples of such relations include verb relations and preposition relations. Consider the sentence:

(S2) John cleaned the grains of wheat.

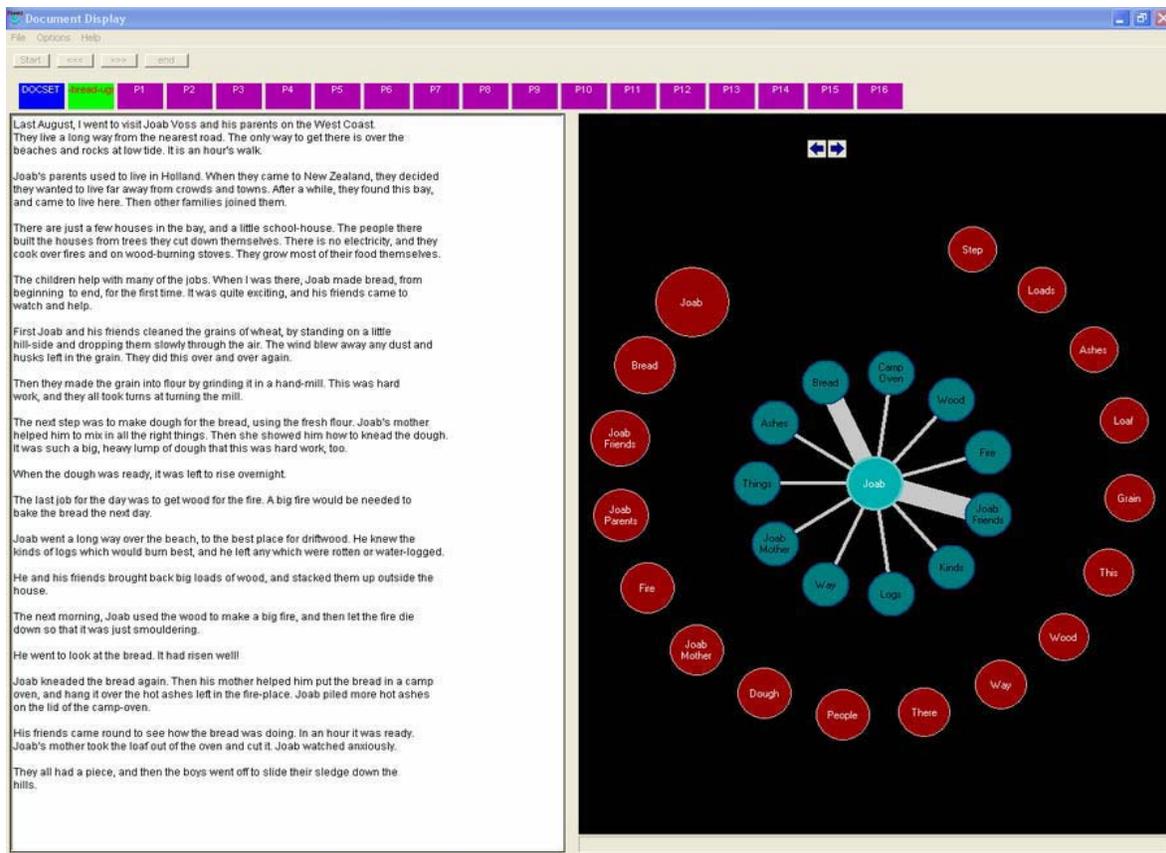
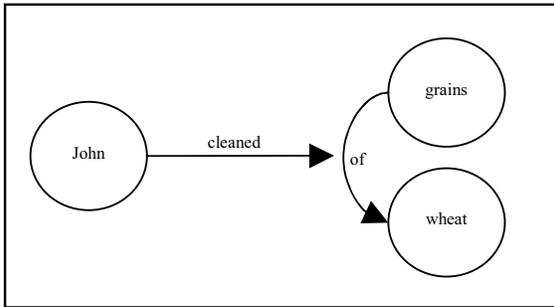


Figure 2: SmartINFO user interface: original text (left) and the Network Overview (right)

At the word level, the following concept network is produced (Fig. 3):



**Figure 3: A simple concept network of S2**

Each noun term is connected to another noun term either via a verb or a preposition, or to a sub-network of more deeply connected noun terms. To visualize the text, users can select a noun term and the system displays all its connections. There are many ways in which this basic approach can be realized.

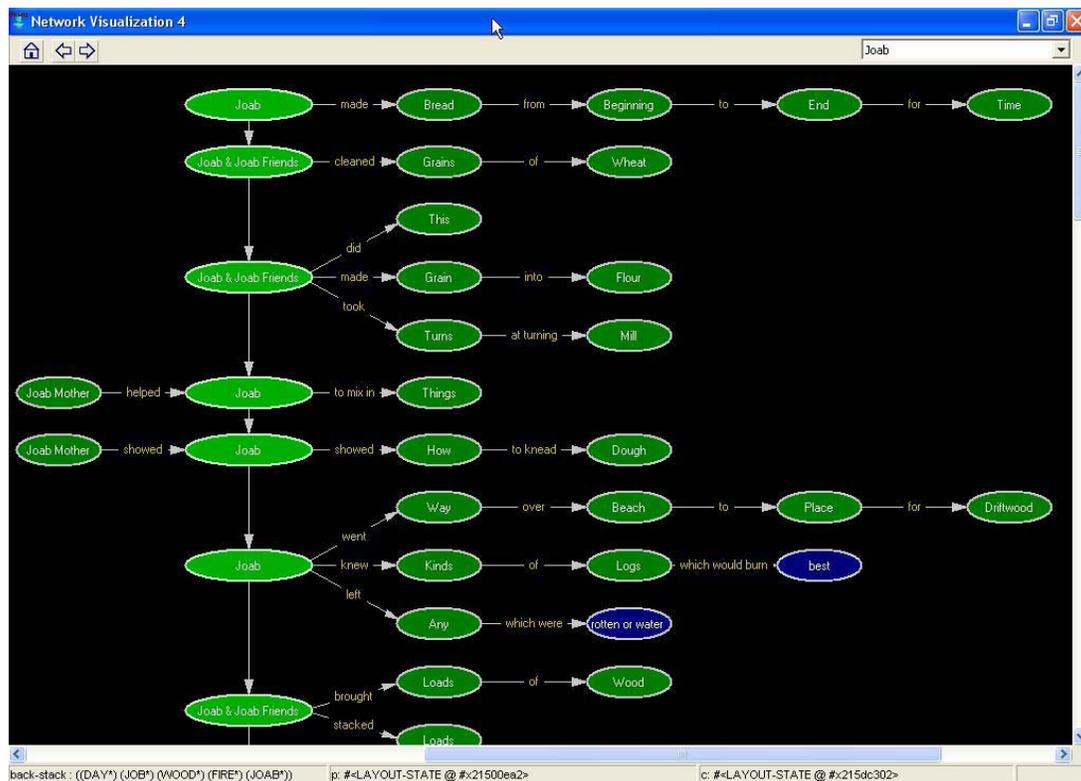
We implemented two views. The first provides an overview of the network (Fig. 2) and the second displays detailed connections of a noun term selected by the user (Fig. 4).

### 3.1. Implementation

The first window allows the user to select and load any number of text documents for processing. This window has two main display frames; the left-hand side frame shows the original document in text format while the right-hand side frame displays an overview of the network computed (Fig. 2). The latter displays two sets of data. First a list of noun terms identified from the text is displayed as a moveable circle of nodes. The size of the displayed node is relative to the frequency of that noun concept appearing in the text. The list is ordered, with the most frequently occurring noun terms displayed first.

Second, in the centre of the network overview frame is a display that shows the interrelatedness between a selected noun term and other noun terms directly connected to it. The thickness of the connecting lines between nodes is relative to how strongly related the two noun terms are in the text.

The overview display highlights the relative importance of noun terms in the text and provides some context surrounding the use of the selected noun term.



**Figure 4: Visualization of term "Joab" using a word-centered representation**

Each displayed node may be clicked to review a visualization of its use in the text. The latter will show all of the relationships that the term is associated with, in the order that they occur in the text (Fig. 4). The sequential order of the text is displayed from top to bottom, with each clause shown as a series of concepts, connected by directed edges, from left to right. The selected concepts from each clause are highlighted and aligned, with adjacent matching concepts merged, where possible

Displayed nodes and connections in the network respond to mouse and keyboard events from the user. For example, a mouse-over event can display extra information about the selected component and a mouse-click event on a display node can set the focus of the network visualization to the selected noun term.

### 3.2. Discussion

In much of the earlier work on text visualization, a word-centered representation is commonly used to represent a text document [1,2,3,4]. It is a useful representation which makes explicit all the connections between terms.

In our system, the resulting display shows what is connected and how it is connected, but the overwhelming detail fails to highlight the key part of each sentence. With each sentence reduced to a linear display of vertices and edges, it is difficult to appreciate which concepts and relationships are most significant. Also, individual concepts have become isolated from the overall context of the document. This loss of context makes it difficult for the user to grasp how the selected relationships relate to each other.

The current implementation does not provide a context to understand the displayed information as a whole. How does each horizontal network relate to the others? Which of the networks are more significant?

### 4. Clause-centered visualization

Our second approach is centered on making explicit clausal relations found in sentences. Each sentence is represented as one or more clauses. With each clause captured as a frame with 3 main slots: subject, verb, and object. An example is shown in Fig. 5.

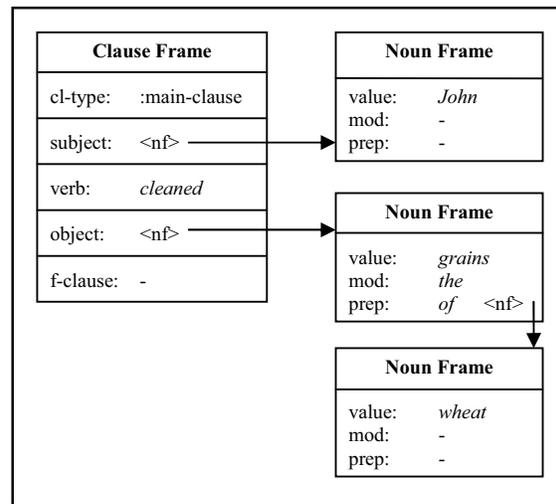


Figure 5: Clause frame representation of S2

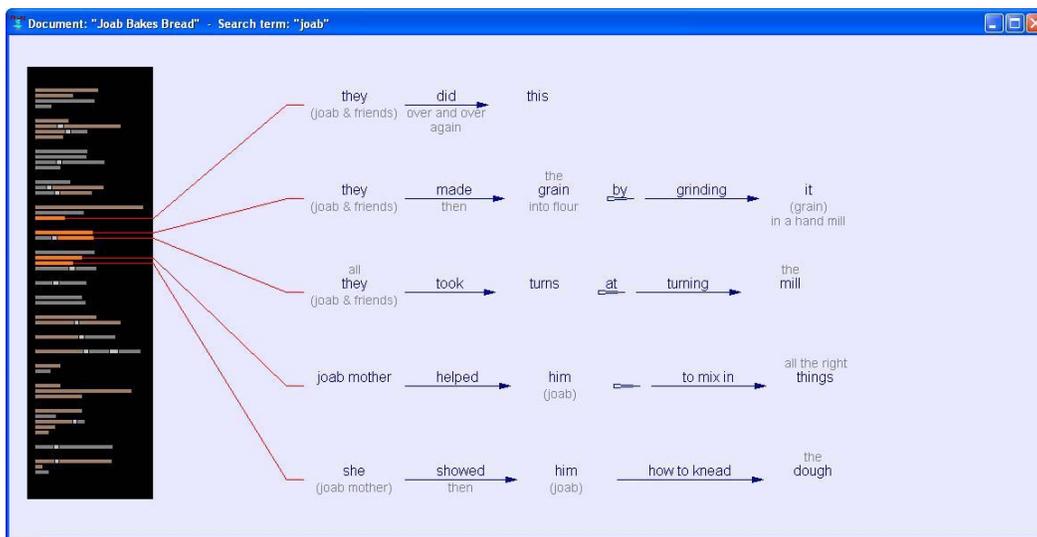


Figure 6: Visualization of meanings using a clause-centered representation, for query "Joab"

A clause can connect to other clauses via the f-clause (following clause) link. Otherwise, a clause connects noun frames together. Additional parts of speech (adverb, adjective, prepositional phrases, etc.) are stored explicitly as part of a noun- or clause-frame.

Using this representation, one has already identified the key part of each sentence (its main clause) and the role of each noun frame in the sentence. The clausal representation provides some means of ranking the information for display by providing a more flexible approach to handling additional information, which can be displayed with more or less emphasis, as required (e.g. prepositional phrase "in a hand mill", Fig. 6).

#### 4.1. Implementation

Fig. 6 shows our new way of displaying meanings using a clausal representation of text. Our initial algorithm displays all clauses found related to the selected term. The user can advance the display, with a mouse-click, to show the next set of matching clauses.

To provide a context for viewing these clauses, a *para-bar* pane is provided on the left side of the display window. This pane shows the structure of the entire document in one view, displaying a picture of each paragraph, sentence and clause.

All clauses that have matched the users query are marked in dull orange, while all unmatched clauses are displayed in grey. Any clause that is currently selected

and displayed in the primary visualization is marked in a brighter orange. A line connects each displayed clause to its corresponding position in the *para-bar* pane so that it is immediately evident to the user how each displayed clause relates to the document as a whole. Equally important, it also shows whether there are any intervening (non-matching) clauses that appear in the text and are not displayed in the current visualization.

In addition to displaying the clauses, related information can now be displayed surrounding the relevant terms. In the implementation, adjectives are displayed on the top of the noun term while other related information is displayed at the bottom.

The idea of displaying extra information has also been extended to displaying anaphors. Although our pronominal anaphor resolution algorithm is highly accurate (approaching a success rate of 80%, see [7]), the system cannot guarantee that a particular anaphor resolution is correct. For the purpose of visualization, replacing an anaphor with an incorrect resolution is worse than not solving the anaphor at all. This is because the user may be given misleading information without being informed that this could be the case. Consequently, in our implementation, anaphors are retained for visualization and the predicted resolution is displayed in brackets below them. Providing maximum information to the user, while indicating the possible uncertainty of that information.

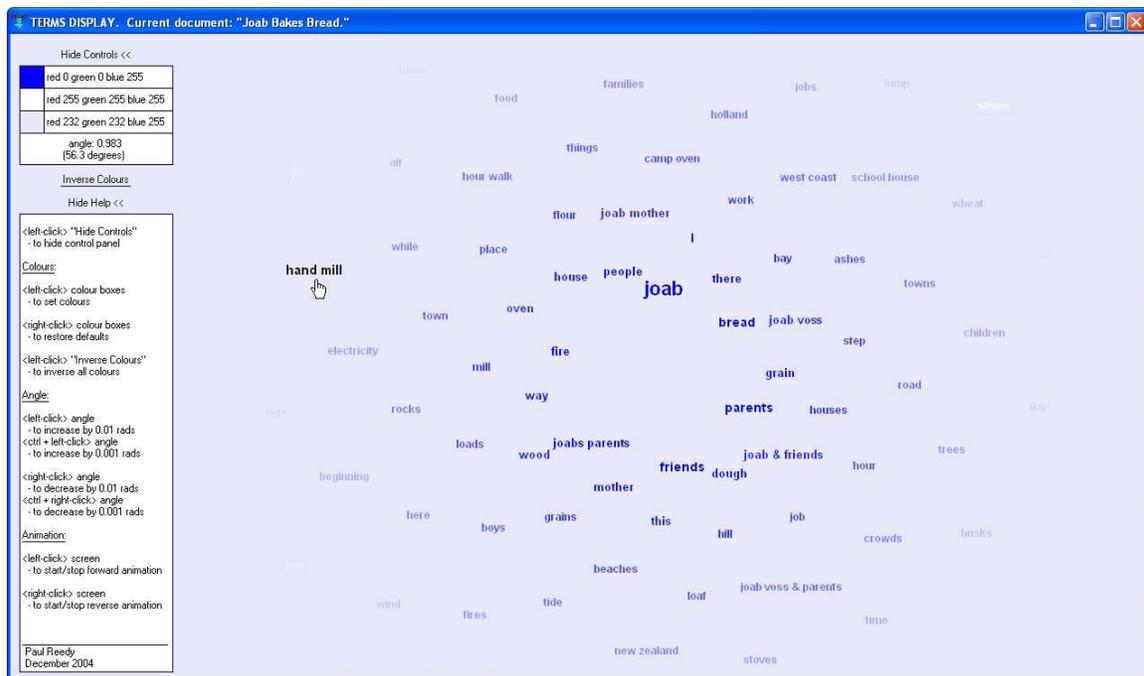


Figure 7: Terms display for whole network, with control-panel showing

Since the text is no longer represented as a network of concepts and relationships, but as a hierarchy of clauses, it is necessary to add another level of processing to index all unique noun terms from each clause. This also enables us to calculate the relative frequencies of each term throughout the document. The original interface to SmartINFO (Fig. 2) is no longer appropriate and a new interface, called Terms Display, is implemented (see Fig. 7). This display shows, on one screen, all the unique noun terms present in the text. The frequency of occurrence for each term is indicated by the size of each display term and reinforced by colour intensity and position.

All the elements in the Terms Display are active to user interaction. A mouse-over event on any term will enlarge and highlight the term (e.g. "hand mill", Fig 7). A mouse-click event will generate the set of all matching clauses for the selected term and launch the corresponding meaning representation display.

## 4.2. Discussion

The clause centered visualization gives the flexibility to differentiate the key part of each sentence from the additional information, for the display. Anaphors are displayed together with the resolved term to give the user an awareness of the possible uncertainty of the information presented. Together with the added context of the para-bar, this display makes the meaning more apparent to the user (compare Fig. 6 with Fig. 4).

## 5. Conclusion

We presented a system for the visualization of the meanings of texts. We discussed two ways of displaying meanings to the users. The first method is word-centered and the second method is clause-centered. The first method is commonly used in many existing text visualization systems. It provides detailed connections between terms. However, we conclude that it is inadequate for conveying the meanings of texts. The second method presents the information gathered, at the appropriate level, and in such a way, as to enable the user to discern the intended meaning of the text. It provides a more interesting framework to advance our work on visualizing meanings of texts.

In the future, our work will focus on two important problems. The first problem is to provide a means of evaluating the significance of the contextual information surrounding each clause. It is important in this visualization task that not every piece of data is displayed all the time. Knowing what not to display is just as important as knowing what to display. The second problem is to provide a richer context for interpreting each clausal output. An example of this

would be to exploit the use of rhetorical structure [8]. We also aim to employ user-testing to validate and further develop our system.

## Acknowledgements

This work is partly funded by a grant from the New Economy Research Fund (NERF) of New Zealand.

Screen shots in this report show visualizations of the text "Joab Bakes Bread" by Ans Westra, published in School Journal, part 2, number 2, 1988. Department of Education, New Zealand.

## References

- [1] A. E. Smith. Automatic Extraction of Semantic Networks from Text using Leximancer. In *Proceeding Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics - Companion Volume*, Edmonton, Alberta, Canada., 2003.
- [2] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *Proceeding The IEEE Information Visualization Symposium*, Atlanta, Georgia, USA., 1995.
- [3] C. Albrecht-Buehler, D. A. Shamma, and B. A. Watson. TextPool: visualizing live text streams. In *Proceeding 10th annual IEEE Symposium on Information Visualization*, Austin, Texas, USA., 2004.
- [4] W. B. Paley. TextArc: Showing Word Frequency and Distribution in Text. In *Proceeding IEEE Symposium on Information Visualization, Interactive Poster Session*, Boston, Massachusetts, USA., 2002.
- [5] W.K. Yeap. Semantics Parsing Revisited or How a Tadpole Could Turn into a Frog. Paper accepted for publication in the 2<sup>nd</sup> Language Technology Conference: Human Language Technologies as a challenge for Computer Science and Linguistics, Poznan, Poland, 2005.
- [6] W.K. Yeap. On Baker's paradox and a new computational theory of language. Paper submitted to the Cognitive Science Conference, Stresa, Italy, 2005.
- [7] H. Ho, K. Min, and W. K. Yeap, Pronominal Anaphora Resolution Using a Shallow Meaning Representation of Sentences, In *Proceedings of the 8<sup>th</sup> Pacific Rim International Conference on AI*, Springer-Verlag, *Lecture Notes on AI*, 3157, 862-871, 2004.
- [8] D. Marcu. The Rhetorical Parsing of Unrestricted Texts: A Surface-Based Approach. *Computational Linguistics*, 26 (3), 395-448, 2000.