# Hierarchical Discourse Parsing based on Similarity Metrics

Ravikiran Vadlapudi, Poornima Malepati, Suman Yelati
International Institute of Information Technology Hyderabad,
Gachibowli Hyderabad India 500019
{*ravikiranv, mpoornima, suman.yelatipg08*}@*research.iiit.ac.in*

## Abstract

Attentional State Theory and Rhetorical Structure Theory are two predominant theories of discourse parsing. Combining these two approaches, in this paper, we describe a novel approach for discourse parsing. The resulting discourse tree structure retains following properties: structure of purpose from Attentional State Theory and relations between sentences from Rhetorical Structure Theory. We demonstrate the utility of our model by constructing a summarization system.

## Keywords

Discourse Parsing, Attentional state theory, Rhetorical structure theory, Coreference, Cohesion, Similarity metrics, Sentence similarity

## 1 Introduction

Discourse parsing is crucial for parsing texts in Natural Language where each sentence has a purpose modelled by relationship with other sentences in the discourse. The main objective of discourse parsing is to generate a valid discourse structure which captures this purpose. Discourse parsing finds its applications in a variety of fields some of which include summarization (focused summaries), dialog systems (language generation) and information retrieval (Question Answering (QA) systems).

A few theories have been proposed for structuring a discourse, two of which are Attentional State Theory (AST) [7] and Rhetorical Structure Theory (RST) [12]. Attentional State Theory (AST) stresses the role of purpose in processing the discourse. It contains three components, namely, a linguistic structure, an intentional structure and an attentional state. The linguistic structure models the structure of a sequence of sentences in a discourse. A sequence of sentences in a discourse are aggregated into discourse segments. Each sentence serves a purpose in a discourse segment and in turn each discourse segment serves a purpose (Discourse Segment Purpose, DSP) with respect to the overall discourse. This structure of purpose of a discourse segment (and in turn the sentence) is modelled by intentional structure. To capture the structure of purpose, discourse segments are related using two relations namely, *dominance* and *satisfaction-precedence*. If the purpose of a discourse segment (DSP1) contributes to the purpose of another discourse segment (DSP2), then, DSP2 *dominates* DSP1. If the purpose of a discourse segment (DSP1) has to be satisfied before the purpose of another discourse segment (DSP2), then, DSP2 *satisfaction-precedes* DSP1. At any given point of time, the discourse segment purpose under focus is kept track of by attentional state.

Rhetorical Structure Theory (RST) represents the structure of a discourse as a hierarchical tree diagram based on the relationship between sentences/text spans (nodes). The relation between these nodes can be of two types: *symmetric* and *asymmetric*. A *symmetric* relation relates two or more nodes labelled nuclei, each of which are equally important in realizing the writer's communicative goals. An *asymmetric* relation relates two nodes, a nucleus and a satellite, the nucleus being more important of the two and the satellite modifying the nucleus based on the particular relation. Due to the hierarchical structure of RST, summarization is fairly simple and efficient with respect to quality of output: the summary generated by omitting satellites of the tree after a certain depth would outline the main points of the text.

Few earlier systems aimed at discourse parsing are: A rule based system RASTA (Rhetorical Structure Theory Analyser) [3] is a component of Microsoft English Grammar that constructs RST analysis of texts. For each relation type, they define a set of rules between nodes, satisfying which, the nodes are related by that relation type. Another rule based system [15], develops a Unified Linguistic Discourse Model (U-LDM) for parsing a discourse. They perform discourse segmentation using discourse semantics and build a discourse tree based on syntactic, semantic and lexical rules. One of the machine learning approaches for discourse parsing [16] learns on discourse annotated corpora. They generate a rhetorical structure tree of a sentence based on the learned models of RST-DT 2002 corpus. Work done in [6] extend LTAG approach to discourse parsing, the D-LTAG. An incremental discourse parsing model has been proposed in [4] which aims at building a rhetorical structure of discourse using veins theory [5]. In [9], an automatic generation of discourse tree has been proposed. The sentences are nodes of the discourse tree and the edges between the sentences define a relation. The relationship type is defined using clue expressions. As per our knowledge, there are no freely available annotated discourse tree banks which makes machine learning approaches infeasible. Rule based systems require tedious analysis of discourse and the rules might not be generic which restricts the domain of usage of the system. Therefore it is necessary to develop an approach which is generic and is un-supervised.

In this paper we combine the ideas of Attentional State Theory and Rhetorical Structure Theory and propose a novel scheme for discourse parsing. We incorporate the features of RST into AST so that we can understand the role of a sentence better and in turn create a better interpretation of the discourse which increases the quality of applications built on it. The discourse tree we build retains following properties: structure of sequence of sentences (linguistic structure) and structure of purpose (intentional structure) from Attentional State Theory and relations between sentences/text spans from Rhetorical Structure Theory. We develop a discourse tree structure based on similarity measures between sentences, identify discourse segments from the discourse tree structure and define relationships between discourse segments and between sentences of discourse segments. We demonstrate the utility of our model by constructing a summarization system. The paper is structured as follows: First we describe our system of discourse parsing in detail in section 2. In section 3 we develop a simple method of summarization based on our system with results. In section 4, we talk about future work and conclude.

# 2 Our Model

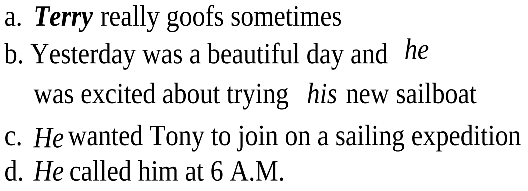
We assume a discourse to be a collection of sub-topics which contribute to the main topic of the discourse. The utterances/sentences which speak about a subtopic are aggregated as a discourse segment (DS). These discourse segments together form the discourse. A Discourse segment serves a purpose called discourse segment purpose (type of contribution to the topic of the discourse). We structure the discourse based on the above assumption by relating sentences which speak about the same topic to form discourse segments. Presently, We define the relationship between utterances/sentences within a discourse segment and the relationship between discourse segments as *dominance*. The relationship specifies the type of contribution to main topic which we refine as a part of future work.

In order to relate sentences which speak about the same topic we use surface information of a sentence and the evidences of cohesion. Two types of cohesion namely *coreference* and *lexical* cohesion prove to be vital evidence to find sentences which speak about a same topic. Coreference expressions refer to a previously introduced entity (center). So the sentence with a reference expression referring to a center in another sentence are said to discuss about the same topic (center). Lexical Cohesion is repetition of the same lexeme or a lexeme which is semantically similar (Hyponyms). Sentences which have similar surface information are also said to discuss about the same topic.

In this section we describe our model of parsing a discourse. We process the given discourse in three phases to progressively build a discourse tree structure similar to rhetorical tree structure[12] ,with one relation type *Dominance*[7], and maintaining the intentional, lexical structure of AST. In the first phase, we use a co-reference module to extract reference expressions (expression which refers to an Object, a Noun Phrase (NP) or a Prepositional Phrase (PP)) and their centers (the Object to which the reference expression refers to). We relate a sentence containing a reference expression to the sentence with the corresponding center as they speak about the same "Object" (topic). In the second phase we relate the remaining sentences using surface information and evidences of lexical cohesion. In the third phase we form discourse segments and find relations among them to generate a tree similar to RST. We now proceed with the description of the system.

## 2.1 First Phase

In the first phase, we use the output of the co-reference module (LingPipe)[1] to create a collection of trees from sentences. Given a discourse text, a co-reference module extracts reference expressions and marks their corresponding centers. For example, in Figure 1 the reference expression *he* refers to *Terry*, the center.



a. ***Terry*** really goofs sometimes
b. Yesterday was a beautiful day and *he* was excited about trying *his* new sailboat
c. *He* wanted Tony to join on a sailing expedition
d. *He* called him at 6 A.M.

**Fig. 1:** *Sample text with centers in bold italics, reference expressions in italics*

From this output of co-reference module we construct trees as follow: at the beginning, we consider each sentence to be a single node tree. Next, we find a parent to each of the sentences using the reference expressions as the basis. For this, consider a discourse of $n$ sentences say $S_1$ to $S_n$. We make a sentence $S_i (i \neq 1)$ the child of $S_{i-1}$ if any of the following hold

- There exists at least one reference expression in $S_i$ with a center in $S_{i-1}$
- There exists at least one reference expression in $S_i$ and one reference expression in $S_{i-1}$ with a common center in some sentence $S_k, 1 \leq k < i-1$

At the end of this phase we have a collection of trees, each node with at most one child. Note that there is a possibility that a sentence, say $S_i$ may not have a reference expression or may not have a center referred to by any reference expression. In this case, the sentence would remain a single-node tree. It is apparent from our construction that every tree now contains a chain of continuous sentences, each tree can be considered a partially built discourse segment. In the next phase, we merge these trees together into a single tree on the basis of lexical cohesion and surface information.

## 2.2 Second Phase

Suppose after the first phase we have a collection of $z$ trees, $T_1$ to $T_z$. We maintain a final tree $T$, initialised to $T_1$, into which we merge all the remaining trees. This merging is done using lexical cohesion and surface information (similarity of content) between nodes (sentences) of candidate trees.

For merging $T_i, i \neq 1$ with $T$, consider the root sentence of $T_i$, say $S_j$. Extract all the Noun Phrases (NPs) and Prepositional Phrases (PPs) of $S_j$ using a parser. We use the standard Stanford parser [8] for parsing. These NPs and PPs constitute *Surface Information.* We exploit the property of lexical cohesion in a discourse by assuming that the NPs of *Subject* semantic role and *Direct Object* semantic role have evidences of lexical cohesion. Therefore, out of all NPs and PPs of $S_j$, we consider the NPs with *Subject* semantic role (*NPSubject*) and *Direct Object* semantic role (*NPObject*) of the main verb. If the main verb is intransitive, we consider only the NP with *Subject* semantic role (*NPSubject*). For merging $T_i$ into $T$, we find the most probable parent of $T_i$ by comparing with the *NPSubject* and *NPObject* (if any) of $S_j$ and the *Surface Information* of $S_{j-1}$, which exists in $T$, and all the ancestors of $S_{j-1}$. Our algorithm for merging $T_i$ with $T$ is as follows:

- Consider a set U = $U_1$, $U_2$,... $U_k$, $U_1$ the surface information of $S_{j-1}, U_2$ to $U_k$ the surface information of ancestors of $S_{j-1}$, $U_k$ the surface information root of $T$.
- Calculate the similarity between $U_1$ with *NPSubject* and *NPObject* (if any) of $S_j$, the process of similarity calculation being discussed later in the paper.
- For $U_1$ pick that entity which has maximum similarity (either with *NPSubject* or with *NPObject*) and define that entity as the topic between $S_j$ and $U_1$.
- Repeat for all $U_i$s and find the maximum of maximum similarities.
- If the maximum is unique and belongs to a node say $S$ of $T$ then we make $S_j$ the child of $S$ and thus merge $T_i$ into $T$.
- If there is more than one occurrence of the maximum, then a decision based on lexical cohesion is not possible.
    - Inorder to find a most probable parent among these maxima, we prioritise the topics based on the importance of that topic. For this purpose, we use TextRank [13], which gives a rank to each topic based on the knowledge from the text. It is a graph based ranking algorithm using Google's Page Rank [2] on text. Out of the maxima, whichever topic has the highest priority, we choose the corresponding sentence as the parent of $S_j$.
    - If there exists a clash of priorities in this level also, then we resort to sentence similarities.

After calculating the similarity between $U_i$ with *NPSubject* and *NPObject*, to avoid irrelevant entities, we pick only those entities/topics which have similarity value above a specific threshold (for better results > 0.9). If there is no topic which has value greater than the threshold, then we consider sentence similarities. While considering sentence similarities, out of all the ancestors of $S_{j-1}$ (including itself), the sentence (node) most similar to $S_j$ is made the parent of $S_j$. For example in Figure 2, $T_1$ and $T_2$ are the first level trees formed from five sentences $S_1$ to $S_5$. We initialise $T_1$ to $T$ and for merging $T_2$ into $T$, we calculate the similarity value between $S_4$ with $S_3$ and all its ancestors. The final similarity measures are shown in the figure, $S_4$ is made the child of $S_2$ as it has the maximum similarity measure.

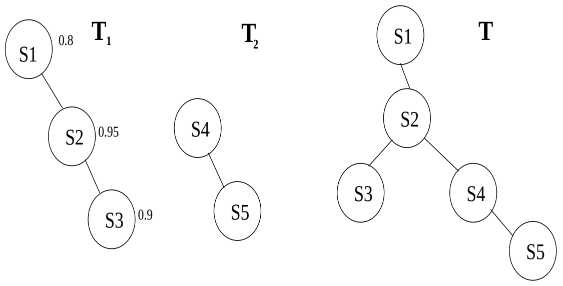


**Fig. 2:** *Tree merging of second phase*

The similarity of our construction to that of Attentional State Theory can now be seen easily. In Attentional State theory, each sentence starts a discourse segment or continues a discourse segment or ends a discourse segment. In our case, each time a sentence $S_j$ attaches itself as the child of $S_k$, it figuratively means that it either continues the topic of $S_k$ or ends the topic of $S_k$. This also means that $S_j$ ends all the topics of siblings of $S_k$. Therefore, from now on for a sentence $S_{j+1}$, there is no need to consider all the nodes of $T$. It would suffice to consider $S_j$ and its ancestors since it cannot continue an already ended topic. It is apparent that we are refining the partial discourse segments constructed in first phase during the second phase. In the next section, we describe the procedure to find similarity between two sentences.

### 2.2.1 Similarity between sentences

Given two sentences $S_1$ and $S_2$, we calculate the similarity measure between these two as follows: Extract the NPs and PPs of $S_1$ and $S_2$, identify the common nouns (NN) and proper nouns (NNP) from each NP and PP. Out of these filter out only those NNs and NNPs whose degree of importance (from TextRank) is above a threshold. For calculating the similarity between these two sentences, we use Dynamic Time Warping Algorithm (DTW) [11] on the filtered set of NNs and NNPs. Dynamic Time Warping Algorithm calculates the sentence similarities using dynamic programming, with a matrix whose rows correspond to the words of first sentence and columns correspond to the words of the second sentence, the similarity score between a word of one sentence and a word of another sentence being the elements of the matrix. In our case the rows correspond to the filtered NNs/NNPs of $S_1$, columns correspond to the filtered NNs/NNPs of $S_2$. For each grid element the following recurrence formula is defined

1. A long time ago, When the Earth was a beautiful young girl floating in space, two powerful kings, the Sun and the Moon, decided that they would rule her.
2. The Sun was strong, bold and hot tempered.
3. He lived in a splendid golden palace, surrounded by thousands of sunbeams that danced around him.
4. The Moon, who lived in a silver palace was a much gentler king, who was waited upon by thousands of twinkling stars.
5. He would sail gently through the sky, casting a soft glowing light on Earth.

**Fig. 3:** *Sample Text*

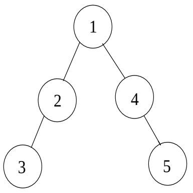


**Fig. 4:** *Final tree after the second phase*

$$D_{dtw}(S_1, S_2) = f(m, n)$$

$$f(i,j) = d(S_{1_i}, S_{2_j}) + min \begin{cases} f(i-1,j) \\ f(i,j-1) \\ f(i-1,j-1) \end{cases}$$

$$f(0,0) = 0, f(i,0) = f(0,j) = \infty$$
$$i \in (0,m), j \in (0,n)$$

Where $m =$ no. of rows, $n =$ no. of columns, $d(a,b)$ the similarity measure between words $a$ and $b$. After the execution of DTW, the normalized value at the last element of the last row gives the similarity measure between $S_1$ and $S_2$. The minimum this value, the maximum the similarity between the sentences. We now describe similarity calculation between a pair of words.

#### 2.2.2 Similarity between words

Several similarity metrics have been proposed so far to calculate the similarity between a pair of words. In [10], the similarity is calculated using the formula

$$Sim_{lch} = -\log \frac{length}{2 * D}$$

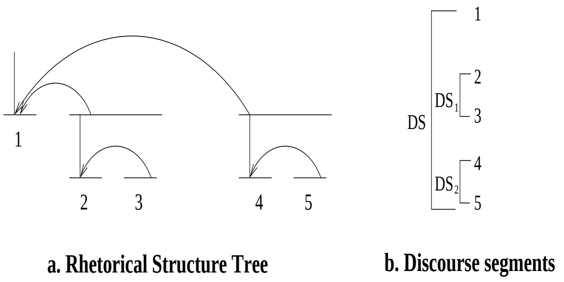
Where length is the length of the shortest path between two concepts using node-counting, and D is the maximum depth of the taxonomy. According to [17], the similarity metric measures the depth of two given concepts in the WordNet taxonomy, and the depth of the least common subsumer (LCS), and combines these figures into a similarity score

$$Sim_{wup} = \frac{2 * depth(LCS)}{depth(concept1) + depth(concept2)}$$

In our paper, we define the similarity score of a pair of words as the average of Synonym similarity and Hypernym similarity. We calculate

$$Sim_w(w_1, w2) = \frac{e^{\alpha d} - 1}{e^{\alpha d} + e^{\beta l} - 2}$$

for Hypernym similarity [11]. $\alpha$ and $\beta$[1] are smoothing factors, $l$ is the shortest path length between $w_1$ and $w_2$; $d$ is the depth of subsumer in the hierarchy semantic nets extracted from WordNet [14]. For Synonym similarity the normalized value of the number of N-gram (N = 3) matchings between the synonyms of two words is taken.



**a. Rhetorical Structure Tree**

**b. Discourse segments**

**Fig. 6:** *Resulting structures from Figure 4*

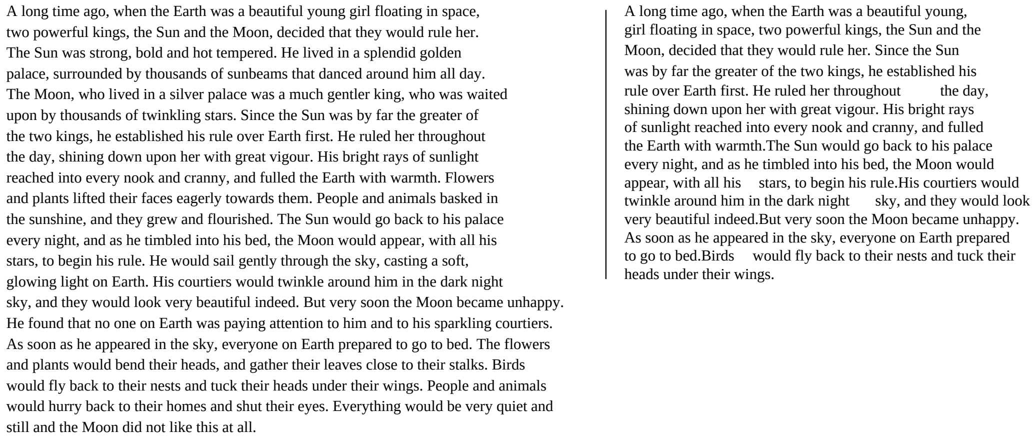
### 2.3 Third Phase

At this stage we have the discourse tree structure where each sentence (node) has a parent (except the root) and each node can have any number of children. We can interpret this tree as follows: The root node starts a new topic. If the root node has more than one children then each of the subtrees of the children in turn contribute to the root node ,but independently. Hence the subtrees rooted at these children can be viewed as separate discourse segments. If the root node has only one child then the root node and the child combined together start a topic. This can be performed at all levels of the tree and a hierarchical discourse segment structure can be built. Each node/sentence that starts a discourse segment has a *dominance* relation to the sentences of that segment.

To make our tree structure similar to that of rhetorical tree structure we identify the nucleus and satellite nodes from the hierarchical discourse segments. Each discourse segment is characterized by a nucleus which is recursively calculated from the sentence(s) starting that segment and the nuclei of the sub-discourse segments of that segment. The nucleus of a segment is the sentence with maximum sum of TextRank values of Noun Phrases (normalized) among the candidate sentences. The remaining sentences are satellites of this sentence.

We demonstrate the algorithm with an example. Figure 3 shows a sample text for which the final tree using the algorithm of second phase described above is shown in Figure 4. Now after the third phase, from the tree in Figure 4, we identify the discourse segments and build RST as shown in Figure 6

[1] $\alpha = 1.0$, $\beta = 0.45$



A long time ago, when the Earth was a beautiful young girl floating in space, two powerful kings, the Sun and the Moon, decided that they would rule her. The Sun was strong, bold and hot tempered. He lived in a splendid golden palace, surrounded by thousands of sunbeams that danced around him all day. The Moon, who lived in a silver palace was a much gentler king, who was waited upon by thousands of twinkling stars. Since the Sun was by far the greater of the two kings, he established his rule over Earth first. He ruled her throughout the day, shining down upon her with great vigour. His bright rays of sunlight reached into every nook and cranny, and fulled the Earth with warmth. Flowers and plants lifted their faces eagerly towards them. People and animals basked in the sunshine, and they grew and flourished. The Sun would go back to his palace every night, and as he timbled into his bed, the Moon would appear, with all his stars, to begin his rule. He would sail gently through the sky, casting a soft, glowing light on Earth. His courtiers would twinkle around him in the dark night sky, and they would look very beautiful indeed. But very soon the Moon became unhappy. He found that no one on Earth was paying attention to him and to his sparkling courtiers. As soon as he appeared in the sky, everyone on Earth prepared to go to bed. The flowers and plants would bend their heads, and gather their leaves close to their stalks. Birds would fly back to their nests and tuck their heads under their wings. People and animals would hurry back to their homes and shut their eyes. Everything would be very quiet and still and the Moon did not like this at all.

A long time ago, when the Earth was a beautiful young, girl floating in space, two powerful kings, the Sun and the Moon, decided that they would rule her. Since the Sun was by far the greater of the two kings, he established his rule over Earth first. He ruled her throughout the day, shining down upon her with great vigour. His bright rays of sunlight reached into every nook and cranny, and fulled the Earth with warmth.The Sun would go back to his palace every night, and as he timbled into his bed, the Moon would appear, with all his stars, to begin his rule.His courtiers would twinkle around him in the dark night sky, and they would look very beautiful indeed.But very soon the Moon became unhappy. As soon as he appeared in the sky, everyone on Earth prepared to go to bed.Birds would fly back to their nests and tuck their heads under their wings.

a. Input text

b. Output summary

**Fig. 5:** *Summarization*

# 3 Summarization and Results

We apply our model of discourse parsing for text summarization. From a given text, we the nuclei of discourse segments upto a certain threshold level which depends on the compression factor.

Figure 5 shows a sample English text and the result of summarization. We plan to test our summarization on a standard corpus. We can improve this model using better summarization techniques on the discourse tree.

# 4 Future Work

Our first task would be to perform a thorough evalution of the current model. Presently our model supports single relation type, *dominance*. We aim at accommodating more relation types into our model. Relation types can be extracted by formulating linguistic rules on cue-phrases such as *therefore, whereas, On the other hand* and so on which would increase the quality of discourse tree further. We can also extend our work to Indian languages such as Hindi which would be a challenging task due to inadequate linguistic resources.

This paper is an outline of system to be developed. The work is still in progress and even though improvements need to be done, we can see that the summary resulting from our rough model is considerably meaningful which in turn signifies the efficacy of our idea.

# References

[1] Alias-i. `http://alias-i.com/lingpipe`, 2008.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine, 1998.

[3] S. Corston-Oliver and S. H. Corston-oliver. Beyond string matching and cue phrases: Improving efficiency and coverage in discourse analysis. In *The AAAI Spring Symposium on Intelligent Text Summarization*, pages 9–15, 1998.

[4] D. Cristea. An incremental discourse parser architecture. *Lecture Notes in Computer Science*, 1835:162–174, 2000.

[5] D. Cristea, N. Ide, and L. Romary. Veins theory: a model of global discourse cohesion and coherence. In *Proceedings of the 17th international conference on Computational linguistics*, pages 281–285, Morristown, NJ, USA, 1998. Association for Computational Linguistics.

[6] K. Forbes, E. Miltsakaki, R. Prasad, A. Sarkar, A. Joshi, B. Webber, A. Joshi, and B. Webber. D-ltag system: Discourse parsing with a lexicalized tree adjoining grammar. *Journal of Logic, Language and Information*, 12:261–279, 2002.

[7] B. J. Grosz and C. L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.

[8] D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[9] S. Kurohashi and M. Nagao. Automatic detection of discourse structure by checking surface information in sentences. In *Proceedings of the 15th conference on Computational linguistics*, pages 1123–1127, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[10] C. Leacock and M. Chodorow. Combining local context with wordnet similarity for word sense identification. In *WordNet: A Lexical Reference System and its Application.* MIT Press, 1998.

[11] X. Liu, Y. Zhou, and R. Zheng. Sentence similarity based on dynamic time warping. In *ICSC '07: Proceedings of the International Conference on Semantic Computing*, pages 250–256, Washington, DC, USA, 2007. IEEE Computer Society.

[12] W. C. Mann and S. A. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[13] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In *Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, 2004.

[14] G. Miller. Wordnet: a lexical database for english, 1995.

[15] L. Polanyi, C. Culy, M. van den Berg, G. L. Thione, and D. Ahn. A rule based approach to discourse parsing. In M. Strube and C. Sidner, editors, *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, pages 108–117, Cambridge, Massachusetts, USA, April 30 - May 1 2004. Association for Computational Linguistics.

[16] R. Soricut and D. Marcu. Sentence level discourse parsing using syntactic and lexical information. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 149–156, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[17] Z. Wu and M. Palmer. Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133 –138, New Mexico State University, Las Cruces, New Mexico, 1994.