Web Document Analysis: How can Natural Language Processing Help in Determining Correct Content Flow?

Hassan Alam, Fuad Rahman[1] and Yuliya Tarnikova
BCL Technologies Inc.
fuad@bcltechnologies.com

Abstract

One of the fundamental questions for document analysis and subsequent automatic re-authoring solutions is the semantic and contextual integrity of the processed document. The problem is particularly severe in web document re-authoring as the segmentation process often creates an array of seemingly unrelated snippets of content without providing any concrete clue to aid the layout analysis process. This paper presents a generic technique based on natural language processing for determining 'semantic relatedness' between segments within a document and applies it to a web page re-authoring problem.

1. Introduction

In web document analysis, document decomposition and subsequent analysis is a very viable solution [1]. In this approach, the web document is initially decomposed into constituent segments exploiting the HTML data structure [2,3]. Once segmented, these segments are then classified into various classes, such as image, text, story (large contiguous chunk of text), titles, side bars, tables, top bars, advertisements and so on [4]. Once the classification of each segment is known, segments of specific classes can be merged using a set of rules. An example of a simple rule might be to merge two story segments that are next to each other in the natural HTML rendering order. Other more complicated rules can also be formulated.

The basic idea is sound and works reasonably well, but it carries all the unpredictability of an empirical system. While merging two segments, the only information available to the merging algorithm is the proximity map and broad content classification. It is not uncommon that sometimes totally unrelated content can easily meet these tests, resulting in the failure of the merging algorithm. This simple scenario provides an idea of the type of problems the web re-authoring applications face in typical conditions. While creating solutions for web page re-authoring, some of the following problems are common:

- How do we determine if two separate web document segments contain related information?
- What is the definition of 'relatedness'?
- If other segments are geometrically embedded within closely related segments, can we determine if this segment is also related to the surrounding segments?
- When a hyperlink is followed and a new page is accessed, how do we know which exact segment within that new page is directly related to the link we just followed?

It is very difficult to answer these questions with a high degree of confidence, as the absence of precise information about the geometric and the linguistic relationships among the candidate segments make it impossible to produce a quantitative measurement about the closeness or relatedness of these segments. This paper has proposed a natural language processing (NLP) based method to determine relationship among different textual segments. The technique is generic and is applicable to any segmented document, but the application area specifically addressed here is web page re-authoring.

2. Natural Language Processing

Computational linguists dealing with *syntax* and *semantics* of languages have long dealt with the problem of making sense of the message conveyed in a narrative. The syntax, in general, is relatively easy to understand and interpret, but the semantics always posed a comparatively complex problem. The problem is compounded by the fact that word usage in any language is full of ambiguity, where the same word may have many senses depending on the *context* of the narrative. Any solution to this problem has to solve some other closely related processing challenges, such as:

- Spell Checking: To verify the integrity of the input.
- Tokenizing: To tag various parts of speech (POS).
- Parsing: To parse and create a representation of the narrative.
- Resolving Anaphora: This is the problem of resolving what a pronoun, or a noun phrase refers to. For example, consider the following discourse: "Fuad is writing a paper for WDA2003. But he

[1] Corresponding author

was very busy". Human readers can easily associate the pronoun "he" referring to "Fuad". However, the underlying process of how this is done is not completely understood.

- Assessing Combined Semantics: Assessing the meaning of individual sentences is one problem, but trying to assess the overall *theme* of a collection of sentences in a narrative is not trivial.

We propose to use these NLP concepts in determining the semantic relationship among different textual segments derived from any document and subsequently use that information in determining the correct content flow. The solution will assume that no geometric information about these segments is available.

3. Lexical Chains

A lexical chain is a sequence of related words in a narrative. It can be composed of adjacent words or sentences or can cover elements from the complete narrative. *Cohesion* is a way of connecting different parts of *text* into a single theme. In other words, this is a list of semantically related words, constructed by the use of *co-reference*, *ellipses* and *conjunctions*. This aims to identify the relationship between words that tend to co-occur in the same lexical *context*. An example might be the relationship between the words "students" and "class" in the sentence: "The students are in class". A lexical chain is a list of words that captures a portion of the cohesive structure of the narrative, and is, by definition, independent of its grammatical structure. Therefore, *context* of a narrative can be computed by creating lexical chains resolving ambiguity and encapsulating the essence of the concept incorporated in the narrative.

Textual cohesion in linguistics [5] is the pre-curser to lexical chaining. The use of lexical chains in determining the structure of texts was first suggested in [6]. Since then, these ideas have been used in many diverse applications, such as summarization [7], information retrieval [8], text segmentation [9], automatic generating of hypertext links [10, indexing [11] and other related areas.

In our implementation, a Commercial-Off-the-Shelf (COTS) spell checker (Sentry Engine [12]) was used. We also used a freely available sentence tokenizer [13] to tag various parts of speech (POS). In addition, we adopted the Minipar parser [14]. We also implemented an anaphora resolver [15].

3.1 Creating Lexical Chains

For every sentence in the narrative, all nouns are extracted. All possible synonym sets are then determined that each noun could be part of. For every synonym set, a lexical chain is created by utilizing a list of words related to these nouns by WordNet relations [16]. Once lexical chains are created, a score for each chain is calculated using the following scoring criterion:

Score = Chain Size * Homogeneity Index

where,
ChainSize = $\sum_{\text{all chain entries (ch(i)) in the text}}$ w(ch(i)); representing how large the chain is, and each member contributing according to how related it is.
w(ch(i)) = relation(ch(i)) / (1 + distance (ch(i)))
relation(ch(i)) = 1, if ch(i) is a synonym,
0.7, if ch(i) is an antonym,
0.4, if ch(i) is a hypernym, holonym or hyponym.
distance(ch(i)) = number of intermediate nodes in the hypernym graph for hypernyms and hyponyms and 0 otherwise.
Homogeneity Index = 1.5 – ($\sum_{\text{all distinct chain entries (ch(i)) in the text}}$ w(ch(i)))/ChainSize; representing how diverse the members of the chain are.

To make sure that there is no duplicate chain and that no two chains overlap, only one lexical chain with highest score is selected for every word and the rest are discarded. Of the remaining chains, "strong chains" are determined by applying the following criterion:

Score >= Average Score + 0.5 * Standard Deviation

3.2 Calculating Relationship between Segments

The previous section discusses how lexical chains for the whole narrative can be constructed. This section discusses how these chains can be used for identifying if the segments of a document are related. We begin by computing if there are enough lexical chains going across these segments to suspect that some of these segments are semantically related. This measure will be called *relatedness factor* and is calculated by:

$$\frac{\sum_{i=0}^{n}\sum_{j=0}^{m}\frac{\alpha(s(i),ch(j))*score(ch(j))}{\min\{size(s(i)),totalsize-size(s(i))\}}}{n},$$

where,
n: the number of segments,
s(i): i^{th} segment,
m: the number of lexical chains,
ch(j): j^{th} chain,
$totalsize$: the number of words on that document,
$size(s(i))$: the number of words in the i^{th} segment,
$score(ch(j))$: the score of the j^{th} chain,
$\alpha(s(i),ch(j))$ ϵ [0, 1]: estimate of how spread out the chain is. This can be calculated in two ways:

***Estimate 1*:**

$$\alpha_1(s(i),ch(j)) = \frac{2*\min\{N_{in}(s(i),ch(j)),N_{out}(s(i),ch(j))\}}{(N_{in}(s(i),ch(j))+N_{out}(s(i),ch(j)))}$$

where,

$N_{in}(s(i),ch(j))$: the number of elements of a chain inside a segment,

$N_{out}(s(i),ch(j))$: the number of elements of a chain outside a segment.

So $\alpha_1(s(i),ch(j))$ is 0 if all elements of a chain are inside a segment, or all outside, and 1 if there are equal numbers of elements on both sides.

Estimate 2:

A similar measure can also be used for estimating how spread out the chain is by using a ratio of nouns belonging to a chain, rather than using absolute counts:

$$\alpha_1(s(i),ch(j)) = \frac{2*\min\{n_{in}(s(i),ch(j)),n_{out}(s(i),ch(j))\}}{(n_{in}(s(i),ch(j))+n_{out}(s(i),ch(j)))}$$

where,

$n_{in}(s(i),ch(j)) = \frac{N_{in/out}(s(i),ch(j))}{N_{in/out}(s(i))}$: the number of nouns inside a segment,

$N_{out}(s(i))$: the number of nouns outside a segment.

For every chain-segment pair, both estimates were used and the highest weight of these two was accepted.

If the *relatedness factor* falls under a threshold (the threshold was empirically set at 1.5), then we consider a document as a set of unrelated or weekly-related segments. Otherwise we identify a beginning segment and an end segment of the main *theme* as the first and last segments that contain an element from a strong chain. This results in identification of segments that are closely related.

4. An Application

Now that we know which segments of a document are closely related, we are in a position to apply it to a practical task. The chosen task is the merging of closely related segments in a web document. **Figure 1** shows an example web document. It also shows how the segmentation algorithm discussed in [1,2,3] creates separate segments in the web page. It is also to be pointed out that the segmentation process relies on the HTML structure of the web page and subsequently do not have any additional information concerning these segments. The natural flow of the segments in most cases does not correspond to the rendered output as seen on the browser window.

Figure 2 shows the relatedness scores and shows that some of the segments are highly related. These segments are denoted by '1' in the figure. **Figure 3** shows how this information is used to reconstruct the segmented document and displayed on a small screen PDA, showing a very concise, yet logically sound representation of the original document.

Figure 1: An example web page

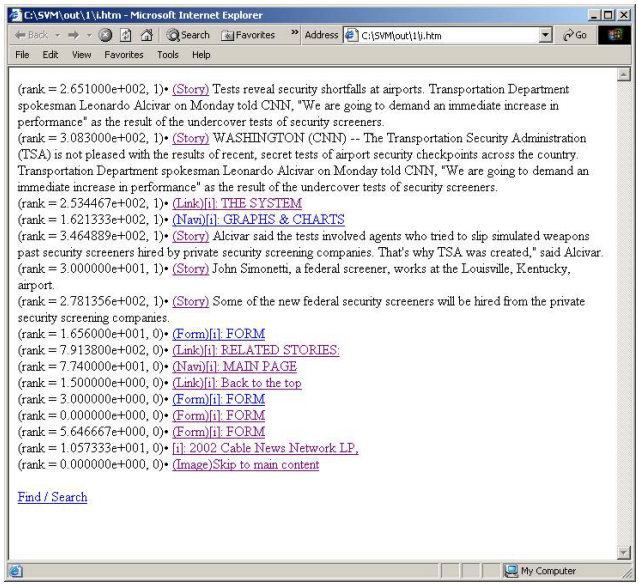

Figure 2: Relatedness scores of the segments

5. Further work

The research on the semantic relatedness reported here is very much a work in progress. One of the drawbacks of the current approach is that only a single main theme can be handled per document. In future we are going to address a more generic solution that can handle documents with multiple themes. Integration of this NLP method in building commercial summarizers and in

aiding existing web page summarization techniques based on structural analysis alone is already well underway. We are also going to explore possible application of this technique in determining the flow of web information between different web pages as the browser loads up new pages following hyperlinks and in aiding geometric web parsers in determining the correct logical layout by complementing geometric information with linguistic coherence.

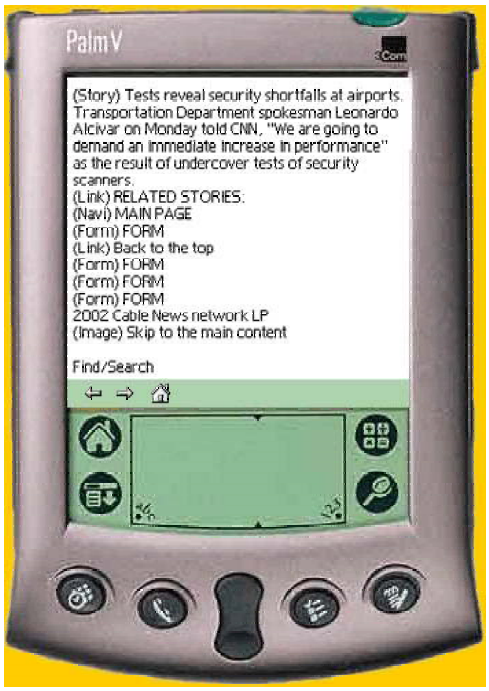

Figure 3: Output displayed on a PDA

6. Conclusion

This paper has presented a novel approach of determining semantic relationship among segments of web documents using lexical chain computation. A novel application of this technique for automatic web page re-authoring is also discussed. In the ICDAR 2003 conference, two related research papers are going to be presented. One will explore the application of lexical chains in building a commercial summarizer capable of summarizing any document [17], and the other will concentrate on a hybrid approach to web page summarization, combining structural and NLP techniques [18].

References

[1] Rahman, A., Alam, H., Hartono, R. and Ariyoshi, K. "Automatic Summarization of Web Content to Smaller Display Devices", 6th Int. Conf. on Document Analysis and Recognition, ICDAR01, pages 1064-1068, 2001.

[2] Rahman, A., Alam H. and Hartono, R. "Content Extraction from HTML Documents". Int. Workshop on Web Document Analysis, WDA01, pp. 7-10, Seattle, USA, Sep., 2001.

[3] Rahman, A., Alam, H. and Hartono, R. "Understanding the Flow of Content in Summarizing HTML Documents". Int. Workshop on Document Layout Interpretation and its Applications, DLIA01, Seattle, USA, Sep., 2001.

[4] Alam, H., Hartono, R. and Rahman, A. "Extraction and Management of Content from Html Documents". Chapter in the book tilted "Web Document Analysis: Challenges and Opportunities". World Scientific Series in Machine Perception and Artificial Intelligence, 2002. In press.

[5] Halliday, M. and Hasan, R. "Cohesion In English", Longman, 1976.

[6] Morris J., and Hirst, G. "Lexical Cohesion, the Thesaurus, and the Structure of Text", Computational Linguistics, Vol 17, No. 1, March 1991, pp. 211-232.

[7] Brunn, M., Chali, Y., and Pinchak. C. "Text Summarization Using Lexical Chains". Work. on Text Summarization. 2001.

[8] Stairmand, M. "A Computational Analysis of Lexical Cohesion with applications in Information Retrieval", Ph.D Thesis, UMIST, 1996.

[9] Boguraev, B. and Neff, M. "Discourse Segmentation in Aid of Document Summarization". In Proceedings of Hawaii Int. Conf. on System Sciences (HICSS-33), Minitrack on Digital Documents Understanding, IEEE. 2000.

[10] Green, S.J., "Automatically Generating Hypertext by Computing Semantic Similarity", Ph.D. Thesis, University of Toronto, 1997.

[11] Kazman, R., Al-Halimi, R., Hunt, W., and Mantei, M. "Four Paradigms for Indexing Video Conferences", IEEE MultiMedia, Vol. 3, No. 1, Spring 1996.

[12] http://www.wintertree-software.com/

[13] Brill E. "A Simple Rule-based Part of Speech Tagger". In Proc. of the 3rd Conference on Applied Natural Language Processing, 1992.

[14] Lin D. "Extracting Collocations from Text Corpora". Workshop on Computational Terminology, Montreal, Canada, pp. 57-63, 1998.

[15] Mitkov R. "The latest in anaphora resolution: going robust, knowledge-poor and multilingual". Procesamiento del Lenguaje Natural, No. 23, 1-7. 1998.

[16] WordNet - A lexical database for the English language. http://www.cogsci.princeton.edu/~wn/.

[17] Alam, H., Kumar, A., Nakamura, M., Rahman, A., Tarnikova, Y. and Wilcox, C. "Structured and Unstructured Document Summarization: Design of a Commercial Summarizer using Lexical Chains. 7th Int. Conf. on Document Analysis and Recognition (ICDAR2003), 2003. In press.

[18] Alam, H., Hartono. R., Kumar, A., Tarnikova, Y., Rahman, A. and Wilcox, C. "Web Page Summarization for Handheld Devices: A Natural Language Approach", submitted to 7th Int. Conf. on Document Analysis and Recognition (ICDAR'03). In press.