An Anaphora-Resolution Based Method ToSegmentText For Knowledge Acquisition

Madhusudanan N[#], B Gurumoorthy, AmareshChakrabarti

[#]*Virtual Reality Laboratory*

Centre for Product Design and Manufacturing

Indian Institute of Science

Bangalore, India

[1]madhu@cpdm.iisc.ernet.in

[2]bgm@cpdm.iisc.ernet.in

[3]ac123@cpdm.iisc.ernet.in

Abstract—**This paper describes the implementation and evaluation ofan algorithm for identifying segments from a piece of naturallanguage text. The aim of the work is to acquire knowledge fromdocuments for the purpose of diagnosing issues in assembly. This is aimed towards connecting two different parts of the Product Lifecycle by means of acquiring and using knowledge. Aircraftis chosen as the domain of application. The first task in theknowledge acquisition process is separation of segments of textthat are relevant to the domain of interest. In an earlier paper, a method has been proposed to carry this out by first segmenting a documentinto coherent sets of sentences and then classifying thesesets (i.e. segments) as to whether they relate to the domain of interestor not. In this paper, we present an implementation of thesegmentation step of this method. The segmentation is based on the identification of common discourse entities shared by physicallyproximal sentences. A challenging part of this process isto resolve anaphora, such as pronouns, in order to identify sentencesthat share their discourse entities. Once the discourse entities are resolved, the resolved set of discourse entities becomesan input to the step of classifying domain-relevant segments. Theprocess of resolving anaphora is difficult. We use the broad approachof Discourse Analysis, and in particular, a tool based onDiscourse Representation Theory, to interpret natural languageas first order logic. However, this tool does not resolve anaphoravery well, as quoted also by the authors of the tool. Hence weperform this step using a specialized anaphora resolution toolkit,and integrate that with the rest of the implementation. The contributionof this paper is a method for segmentation of relatedsentences by weaving together sentences with shared anaphoraand proximity of position. An implementation of this method is described in the paper and results of testing it with a documentrelated to aircraft assembly are shown.**

***Keywords*— knowledge acquisition, natural language understanding, segmentation, discourse analysis, semantic interpretation, first order logic, anaphora resolution**

I. INTRODUCTION

Documents provide a rich resource for knowledge, with the need for storage and retrieval of knowledge during various phases of a product's lifecycle. Knowledge from documents in one part of the lifecycle can be used to avoid potential complications in other parts of the lifecycle. Such knowledge may be about the processes [1] or meta-knowledge to seek further help [2]. There have been efforts to extract such textual knowledge from documents in the past (e.g. [3]).

In the larger context of the research reported in this paper, the aim is to predict assembly issues using knowledge extracted from documents. The chosen domain of application is the assembly of aircraft structures. In order to acquire knowledge from documents, we have to first identify which parts of the documents are of interest from the point of view of knowledge acquisition. These are segments of the document which pertain to our domain of interest. Hence the topic(s) of these segments is a key criterion for such identification.

Any document is not necessarily composed of content about a single topic. The focus of a document may shift between sections, subsections, paragraphs, and possibly, within paragraphs. Also, it is possible that there may be many documentswhere the contents of interest are only a small portion of the entire document. Hence, in order to avoid wasteful processing of the entire document, the first step in this process is to identify segments of documents that pertain to the domain of interest. Thisproblem is further broken down into two broad steps, namely,

1. identification of coherent (sometimes referred to as cohesive in e.g. [4], [5]; However a distinction is made between the two in [6]) sentences that form a meaningful segment in thetext
2. classification of such segments to retain only those that are related to the domain of interest, namely, aircraft assembly

The work reported in this paper is part of the first step above, namely segmentation. A procedure has already been proposed previously[7];this paper focusesspecifically onthe details of implementation and validation of this procedure. In the following sections, we first discuss the need for segmentation,and a short discussion on Discourse Analysis is then provided.The proposed algorithm is then briefly introduced. The detailed procedure of implementation is then discussed, before the implementation of the segmentation is presented. Finally validation of the implementation is presented and future work is outlined.

II. ROLE OF DISCOURSE ANALYSIS

In this research, a document is treated as a one-way discoursebetween the author(s) and the reader. The idea is to use the theory and tools for Discourse Analysis [8] to understand a document and extract required knowledge from the document. The theory of discourse analysis helps us to understand the structure and meaning of an exchange of natural language text. In particular, Discourse Representation Theory is a theory that models discourse as a combination of discourse entities and conditions, and represents themas a structure called Discourse Representation Structure (DRS). The tool-set related to Discourse Analysis (discussed in SectionIV.D) can be used to interpret natural language sentences as a structure containing the Discourse Entities (DE) and the conditions in First Order Logic.

Figure 1 A sample output from JavaRAP

Towards achieving the two steps discussed in Section I, an algorithm has been previously proposed [7],to identify coherent segments of text and classify the segments based on their relation to the domain. The algorithm is repeated in Section III below.

A. Discourse Segmentation

As mentioned in Section II, a document is treated here as a discourse. There is a need to identify sections of a document that discuss one or more topics in a coherent manner. Such sections may be in between paragraphs of text or across paragraphs.

Paragraphs are usually seen as a natural means to maintain focus. However it may not be always the case, as in itemizations. The approach used here is based on understanding the semantic content of the text. The basis for segmentation is the list of discourse entities. The expectation is that sentences which are close to each other in the text and share some portion of their discourse entity list must be related to the same topic(s) of interest. Any variation in the topic of discussion must also be reflected as a change in the set of discourse entities in the text.

B. Anaphora resolution

In order to identify whether sentences share discourse entities or not, two aspects were considered important. The first was to extract the discourse entities from natural language text in each sentence. This step was performed in this research using a freely available set of tools for interpreting natural language texts as Discourse Representation Structures (DRS). The tools are collectively called Boxer and C&C Tools [9, 10]. The second step was to resolve discourse entities that are not resolved by DRS interpretation alone. For this, parts of the text called anaphora had to be resolved. Anaphoras are back-references in a given sentence to entities in previous sentences. Anaphora resolution helps to build the list of discourse entities further. We take the view that if a sentence refers back to an entity in a previous sentence in the discourse, the two sentences must be part of the same segment.

C. Tool for anaphora resolution

It was initially assumed that the set of tools used for semantic interpretation could also perform anaphora resolution at a satisfactory level. However, this was found to be not the case [11]. Further, the tool-set's inherent Segmented DRT could not be utilized due to implementation issues. Hence for the purpose of anaphora-resolution, a different tool, namely JavaRAP [12, 13] was used. JavaRAPis a public Java-based implementation of the Resolution of Anaphora Procedure (RAP). It takes as input an unprocessed natural language text and outputs the resolved anaphora in a text format (see Figure 1). It partitions the text into sentences and tokenized words. Each of the sentences and the words in the sentences has an index assigned to them. These indices are used by JavaRAP to refer to particular words while indicating the results of anaphora resolution. Thus JavaRAP is useful for automatically resolving anaphoric references. Section IV.B contains examples and more details of how JavaRAP is used in this work.

III. PROPOSED METHOD FOR SEGMENTATION AND CLASSIFICATION

The proposed method for identification of text segments and classification is as follows [7]:

1. Given text from a document, tokenize it into sentences
2. Resolve anaphora and pronouns for each sentence to obtain a DE list for that sentence
3. Segment the sentences which are both close in proximity and share parts of their DE list
4. Once the segments are recognized and marked, compare entities in the DE list to determine to how many of them relate to the assembly domain. The basis for comparison here are the

set of terms (and possible terms close to these) from one or more assembly ontologies

5. If the semantic similarity for a segment is greater than a threshold, then classify that segment as being related to assembly

For this proposed method, the implementation described in this paper is for the segmentation part (The first three steps in the above method). The procedure for implementation is described in greater detail in the following section.

IV. IMPLEMENTATION

The implementation of the above procedure is explained in this section. The detailed procedure is shown in Figure 2.Details of each step of implementation are further explained in the subsections below.

A. Assigning sentence indices

As an initial step, we needed an identifier to mark sentences. It would have been possible to simply tokenize the input text on the basis of the end of individual sentences. However, since JavaRAP uses indices of sentences to point to anaphora and their respective discourse referents, it is imperative that we use the same indices. The indices refer to not only the sentences, but also have an index to each word of each sentence. The JavaRAP Sentence Splitter utility is used to carry out this activity. It assigns indices to sentences starting from the number zero.

B. Anaphora Resolution

Once we have indexed sentences and words of our text, the next step of implementation is to resolve the anaphora in text. As mentioned in the previous sections, we use the JavaRAP tool to perform anaphora resolution. We input the text directly into JavaRAP, and the results of anaphora resolution resemble those shown in Figure 1. The indices that we have separately identified in the previous step are used in combination with JavaRAP results to identify and connect anaphora and their referents.

C. Replacing Anaphora

After resolution of anaphora, we decided to replace the anaphora in sentences. This is useful when we interpret sentences. For example, in the sentences

"*Riveting is a complicated process. It involves many parts and tools.*",

When we replace the anaphora "it" with its referent, the second sentence now reads as

"*Riveting involves many parts and tools.*"

The use of doing this is that during semantic interpretation, instead of the anaphora, their actual referents are interpreted. However, the major challenge in replacing anaphora with their referents is that of the form of words to be replaced. For example, consider the sentence

"*Manufacturing is the process of realizing products from their design.*"

Replacing the anaphora results in

"*Manufacturing is the process of realizing products from products design.*"

which is not the desired form.

Thus, the replacement of anaphora also needs to be done with the proper form of word, rather than just the word. The other challenge is that of phrasal anaphora. Theseare not handled in our implementation, since it requires further work on the correct morphological form (also called morphosyntactic information) of the word.

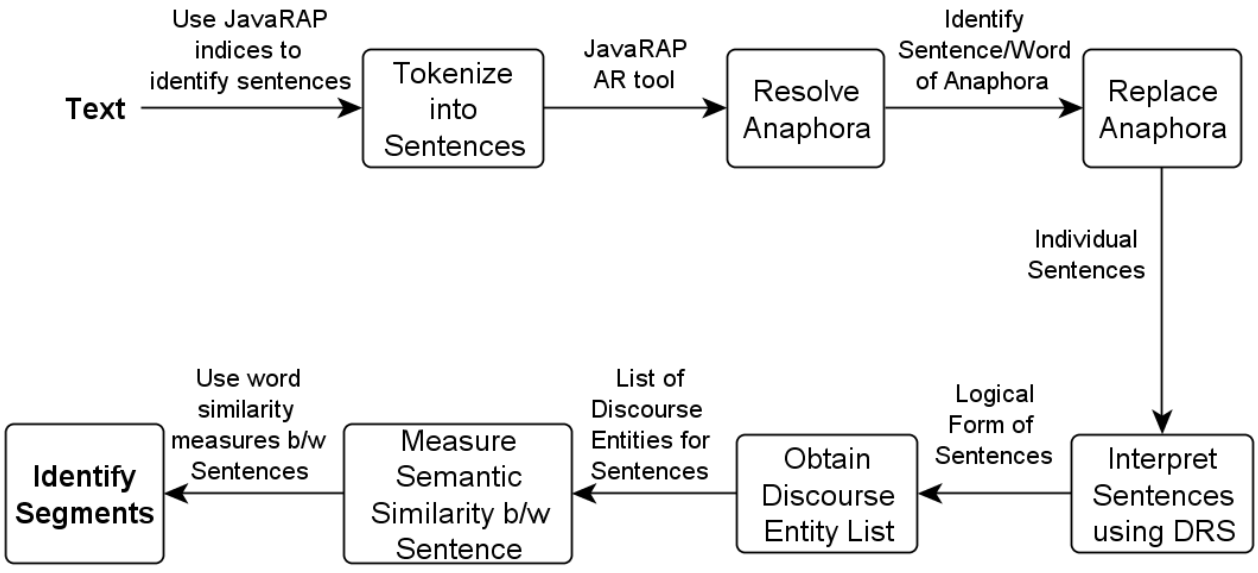

Figure 2 : Overview of the implementation of segmentation algorithm

```
Riveting is the process of joining two plates of metal using a pin
called a rivet.

([p1,x1,x2],[n_process(x2),
prop(p1,([e1,e3,e2,x3,x4,x5,x6,x7,x8],[(x1 = x2), for(e1,e3),
patient(e3,x6), agent(e3,x3), v_use(e3), theme(e2,x8),
recipient(e2,x7), v_call(e2), n_rivet.(x8), (x6 = x7), n_pin(x6),
patient(e1,x4), agent(e1,x3), v_join(e1), of(x4,x5), n_metal(x5),
n_plate(x4), card_eq_2(x4), (x2 = x3)])), n_riveting(x1)])
```

Figure 3 An example of a DRS output by Boxer and C&C tools

D. Interpretation of sentences

After resolving and replacing anaphora in sentences the next step is to interpret these sentences to get their meaning in first order logic. For this, we make use of Discourse Representation Theory[8], in particular, Discourse Representation Structure (DRS).The corresponding tool for interpreting text is the C&C and Boxer toolset. We make use of the Natural Language Tool-Kit (NLTK)[14]in Python programming language, which has interfaces to the above tools built into it. The function of the tool is to take in plain English text and output a DRS interpretation of the text.An interpretation would resemble that shown in Figure 3.

A DRS interpretation has two components

- Discourse Referents: These are the objects (or entities) in the particular Discourse Representation Structure. They are represented by variables in Boxer, and explained in turn using Discourse conditions. They can also contain pronouns, which are in turn resolved using equality (identity) assignment.
- Discourse Conditions: These are conditions in first order logic that represent the relations between discourse entities in the sentence. The conditions are predicates that convey the meaning of these sentences. They can also contain statements that convey the resolution of pronouns. Due to the recursive nature of DRSs these conditions may also contain other DRSs.

When the DRS tool outputs its interpretation, it is in a different form than the 'boxed' diagram shown in Figure 4.

E. Obtaining Discourse Entity List

The purpose of semantic interpretation in this paper is to obtain the list of Discourse Entities. Once the DRS interpretation of texts is obtained, it is directly possible to get the list of discourse referents for each sentence. This is achieved through one of the methods in the NLTK-Boxer interface described above. In the Figure 4 shown above, the discourse referents would be the variables p1, x1, x2. Further on, it is possible to get the discourse entities by going through the discourse conditions that are about an object. Such conditions are predicates of the form n_xxxxx(), ne_nam_xxxx(), and so on. The difference between such an approach and Part-Of-Speech (POS) tagging is that there is a lot more information in this approach than just the POS tags – such as the relation of the object to other objects and events.

An example of the list of discourse entities for a sentence is shown in Figure 5.

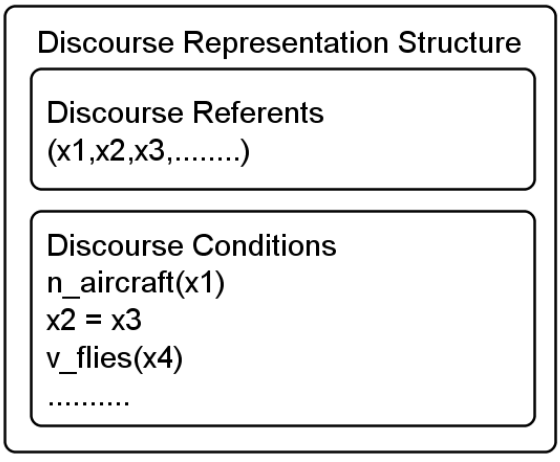

Fig. 4 A generic DRS interpretation of a text

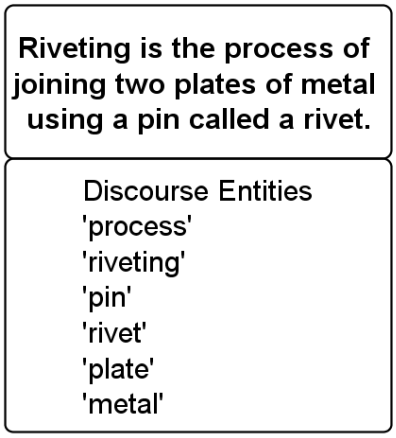

Figure 5 Discourse Entity List for a Sentence

F. Measure Semantic Similarity Between Sentences

Once discourse entities are obtained for every sentence, the next step is to compare how similarity between sentences varies. To do so, we needed a measure of how similar or different two sentences are to each other. For this we propose a measure based on the semantic similarity between words. Examples of such similarity measures between individual words in WordNet are Jiang-Conrath similarity, Lin Similarity and path similarity.

The challenge here is to arrive at a measure of similarity between sets of words based on similarity between words. These sets of words are the Discourse Entity lists.

For example, consider the two lists

[*'quantity'*, *'part'*, *'riveting'*, *'tools'*], and,

[*'surfaces'*, *'rivet'*, *'problem'*, *'access'*]

We can average the semantic similarity measures for all the words from the first list to the second list. However, for each word from the list, there are many ways to measure for each word itself. Some of them are:

- Use the maximum among pairwise comparisons of every word
- Use the minimum among pairwise comparisons
- Average all pairwise comparisons

Depending on which option we choose from the above, the averaging method for the overall score would also differ. In the first two cases, they would have to be averaged over the number of elements of the first set. In the third case, the average must be taken over the elements of both sets.

The Table I shows how the above calculations for each word-pair are carried out in the implementation.

For the above lists of words, the overall similarities between the lists are

- Average Max: 0.4401
- Average Min: 0.0
- Average of Averages: 0.0381

TABLE I
EXAMPLE VALUES OF SIMILARITY

Word	'surfaces'	'rivet'	'problem'	'access'	Min	Max	Average
'quantity'	0.0000	0.0000	0.2754	0.0589	0.0000	0.2754	0.0836
'part'	0.0000	0.0000	0.1052	0.0802	0.0000	0.1052	0.0463
'riveting'	0.2297	1.0000	0.0000	0.0000	0.0000	1.0000	0.3074
'tools'	0.3799	0.3102	0.0000	0.0000	0.0000	0.3799	0.1725
Average					**0.0000**	**0.4401**	**0.0381**

G. Identification of segments from values of similarity

After getting the values of similarity between sentences, these values must be used to identify segment boundaries where such values have large differences. As to the strategy that has to be adopted for doing so, there is no single method that can perform this. For example, Hearst[15] used a method of segmentation based on change of slope and a cut-off value to do so.

We identify a method of finding such segments based on a comparative study of a manual reading exercise versus our calculated values. This is discussed in the following section.

V. VALIDATION OF THE IMPLEMENTATION

In order to validate the implementation of this method, we do a comparative study. First we run the Python program with a sample text to find out similarity scores between sentences. The scores now have to be collated to represent a single measure of how the meaning varies throughout the text. The sample text had 31 sentences in it. It consisted mostly of text about riveting. At a couple of places, sentences of completely different context (two sentences about sports and three about employee satisfaction) were inserted. Feedback about how similar are adjacent sentences were obtained from 11 subjects. This feedback is shown in Figure 6. From the feedback from subjects, we marked segments that were considered stark changes in the topic of discussion. For this there were two factors that were considered:

- A drastic decrease in the similarity scores between adjacent sentences
- A low/very low score of similarity between adjacent sentences

We considered two gradations of both the above scores –a major change, and a minor change in each of these.

Since it was not apparent which score of similarity would reflect a change in context, we had to try with various options. As discussed in the previous section, there are Average, Min and Max values of inter-sentence similarity.

Comparing the plots with the subjects' feedback, there is no single measure that corresponds at most locations. This is because each of these measures behave differently. To understand this, consider the analogy of the two word lists with two clusters. The average of two words roughly

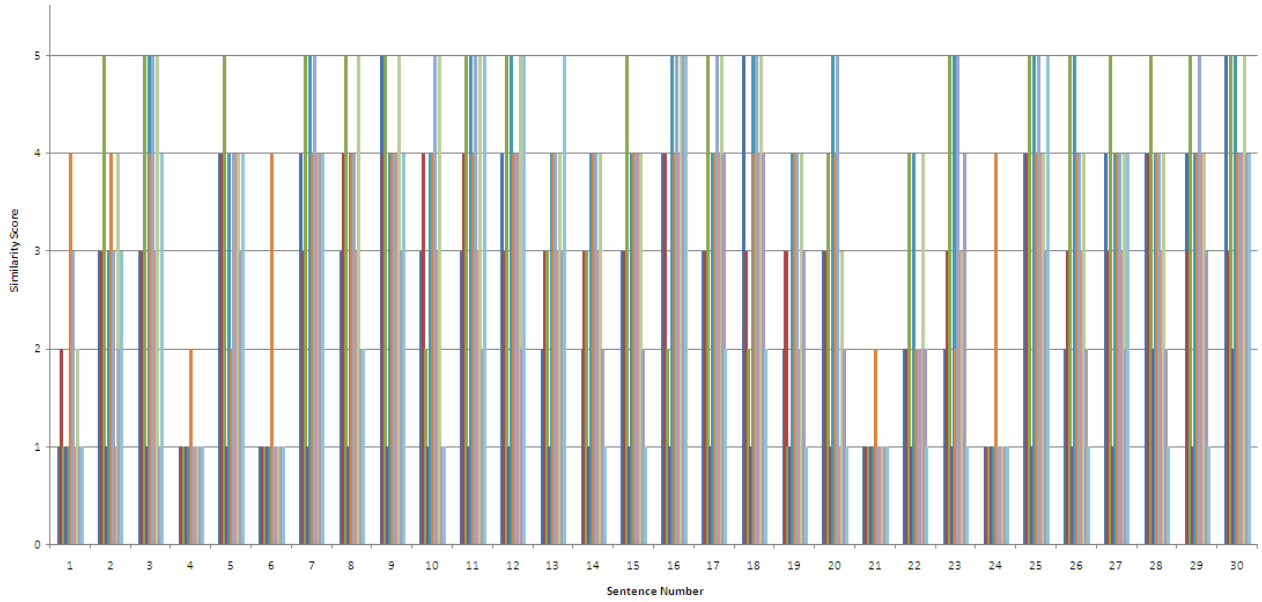

Figure 6 Feedback obtained from eleven subjects for a total of total of 30 adjacent sentence-pair-similarity

represents the centre of the clusters, the min and max would represent the two closest and farthest points respectively. Our purpose here is to find separation between two words sets, given that these word sets may have intersecting elements.

It was decided to combine all three measures into a single measure to use all their qualities. The proposed measure normalizes each of the changes in average, min and max scores of every sentence pair and sums them up. The normalization (to 1) is done so as to give equal weights to each of the above. The measure for a given Sentence *i* would then be,

This is plotted against each of the sentences, as shown in Figure 7. Out of the four minimum points shown in the plot and marked in red circles three are the same major points indicated by a majority of subjects (3, 5, 20, 23). A minor point indicated by subjects (22) also has a minimum.

However, we still need to explain the other minimum points in the plot, which havenot been marked by the subjects.

- The point 10 has a minimum value since the first of the two sentences is a section heading having one word only. It is possible that this single word (‘*Riveting’*) is responsible for small values of Average and Min values, while the Max value is high.
- The other fourmin values are points 12, 14, 16 and 18. Point 12 has also been marked as a minor point, and also the word ‘*sealed’* has not been taken for comparison for the program since it is a verb. Similarly for the point 14, ‘riveted’ was a verb and not considered for comparison. Point 16 also has some spurious considerations, such as not considering ‘*rivet’* and ‘*sheet-metal*’, but considering words like ‘*such’* and ‘*thing’*.Hence correct similarity values have not been calculated at this place between the following sentences: ‘*It is possible to rivet plates of large thickness, such as in bridges*’ and ‘*It is also possible to rivet sheet-metal, as is the case in aircraft construction.*’. Point 18 has been indicated as a minor point (by 5 subjects for individual scores and 2 subjects for difference).Similarly Point 24 has low scores, since entries for the word ‘salary’ and ‘them’ were not found. Point 29 corresponds to sentence pairs 29-30 and 30-31. The dip in value is interesting since, both 30 and 31 are related to 29 only (it is similar to an itemization). Hence the value between Sentence 30 and Sentence 31 is not high, although subjects have marked it so.

Hence it is possible for us to make the following observations:

- A large decrease in slope indicates the presence of a change of topic;
- Titles of sections, when included, create anomalies;
- A combination of the difference values of average, min and max values seemsusefulindistinguishing segments.

The followingare some issues and possible improvements:

- Some words do not have an equivalent synset in the WordNet collection. *Riveting* can be either a noun or a verb. We have currently chosen the closest noun of another form of the word.
- Verbs couldalso be counted for semantic similarity, but VerbNet has to be usedfor this e.g.*manufacturing* is a verb only – hence we use its closest WordNet entry.
- A linear change in context is assumed. This is not always the case, however.

- We are yet to enhance similarity values between sentences based on an anaphoric link.

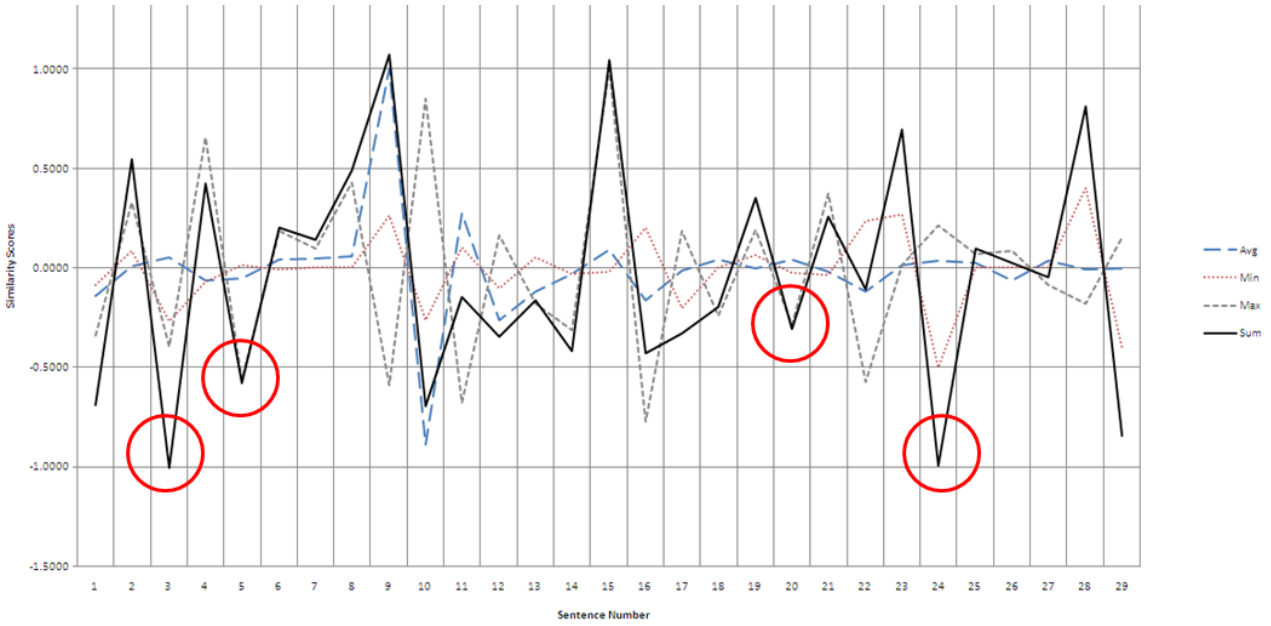

Figure 7 Calculated values of similarity using the implementation

VI. CONCLUSIONS AND FUTURE WORK

An anaphora-based method for identifying text segments has been implemented and evaluated. The evaluation showsa good agreement with the feedback from human subjects. Since the method involves understanding of text, it is a useful prospect for this research. This is becausemost of the downstream activities of knowledge acquisition canbe integrated with the segmentation step. The common threadamong the segmentation and knowledge acquisition activities isthe semantic interpretation of the text.

Since we baseour implementation on available tools, the overall efficiency is heavily dependent on that of these tools. The similarity measure needs to be validated further and improved asnecessary. We are also yet to cover other aspects of discourse entities that may be indicated as verbs or adjectives in the logical form. Including them might also improve the semantic measurements. Another aspect yet to be coveredis that of anaphoric phrases. Although they have been indicated byJavaRAP, we have not enabled the implementation toprocess such phrases.

With respect to evaluation, it is necessary to understandthe feasibility of using segmentation evaluation metrics like quantitative metrics such as the P_k metric [16] for our evaluation.

In the future we intend to connect this implementation with the classificationstep. The integrated implementationwould then serve as input to the actual knowledge acquisition.

ACKNOWLEDGMENT

The authors wish to thank all the subjects from the Centre for Product Design and Manufacturing, Indian Institute of Science who participated in the study.

The authors are also grateful to the various authors of different tools used in this research, such as Johan Bos for Boxer and C&C tools, Long Qiu for JavaRAP, DanGarrette for the NLTK implementation of Boxer interface. Without these tools being available in an open form it would have been impossible to create this implementation in its current form.

This work is performed with funding from The Boeing Company, under SID Project PC 36030.

REFERENCES

[1] Herrmann, J. W., Cooper, J., Gupta, S. K., Hayes, C. C.,Ishii, K., Kazmer, D., Sandborn, P. A., and Wood, W. H., 2004. “New directions in design for manufacturing”. In ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. 853–861.

[2] Sim, Y.-W., Crowder, R., and Wills, G., 2006. “Expert finding by capturing organisational knowledge from legacy documents”.

[3] Lee, T. Y., 2009. “Adaptive text extraction for new productdevelopment”. In ASME 2009 International DesignEngineering Technical Conferences and Computers and Informationin Engineering Conference, American Society ofMechanical Engineers, pp. 769–778.

[4] Foltz, P. W., Kintsch, W., and Landauer, T. K., 1998. “The measurement of textual coherence with latent semantic analysis”. Discourse processes, 25(2-3), pp. 285–307.

[5] Giora, R., 1983. “Segmentation and segment cohesion: On the thematic organization of the text”. Text, 3(2), pp. 155– 181.

[6] Morris, J., and Hirst, G., 1991. “Lexical cohesion computed by thesaural relations as an indicator of the structure of text”. Computational linguistics, 17(1), pp. 21–48.

[7] Madhusudanan, N., Gurumoorthy, B., and Chakrabarti, A., 2014. “Segregating discourse segments from engineering documents for knowledge acquisition”. In Product Lifecycle Management for a Global Market. Springer, pp. 417–426.

[8] Allen, J., 1995. Natural Language Understanding (2NdEd.). Benjamin-Cummings Publishing Co., Inc., RedwoodCity, CA, USA.

[9] Curran, J. R., Clark, S., and Bos, J., 2007. “Linguistically motivated large-scale nlp with c&c and boxer”. In Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, Association for Computational Linguistics, pp. 33–36.

[10] Boxer and C&C Tools. http://svn.ask.it.usyd.edu.au/trac/candc/wiki/boxer. [Online; accessed January-2015].

[11] Bos, J., 2008. “Wide-coverage semantic analysis with boxer”. In Proceedings of the 2008 Conference on Semantics in Text Processing, Association for Computational Linguistics, pp. 277–286.

[12] Qiu, Long, Min-Yen Kan, and Tat-Seng Chua. "A public reference implementation of the rap anaphora resolution algorithm." *arXiv preprint cs/0406031* (2004).

[13] http://aye.comp.nus.edu.sg/~qiu/NLPTools/JavaRAP.html

[14] http://www.nltk.org/

[15] Hearst, M. A., 1994. “Multi-paragraph segmentation of expository text”. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp. 9–16.

[16] Pevzner, L., and Hearst, M. A., 2002. “A critique and improvement of an evaluation metric for text segmentation”. Computational Linguistics, 28(1), pp. 19–36.