A Corpus-Based Lexical Network for Measuring Topic Alignment of Forum Messages

Rodolfo C. Raga, Jr. Jennifer D. Raga
Computer Science Department
Jose Rizal University
80 Shaw Boulevard, Mandaluyong City, Philippines
{jundy.raga, jennie.raga}@gmail.com

ABSTRACT

This paper presents a proposed methodology for the analysis of the topical alignment of forum messages. Aspects of the semantic representation provided by Random Indexing are combined with some statistical properties of words in text to generate a lexical network which we believe represents the topical structure of a text. This approach allows assessment of the degree to which the topic of a message is relevant to the topic of an external reference document. Analysis is done using procedures adopted from the Lesk word overlap algorithm. Application of this approach to messages in a discussion transcript with manually assigned topical alignment values indicate that the performance of the system is still below the average inter-rater reliability of human coders, however, its performance can still be enhanced by exploiting other resources. We also believe that the results obtained in the experiments have raised a number of points which may be worthy of future investigation. We provide some findings and suggestions from the literature that we deem supports our conjecture.

Keywords

Forum Message Analysis, Random Indexing, Lesk Algorithm

1. INTRODUCTION

Forums or online threaded discussion boards are a popular form of web-based computer-mediated communication. In recent decades, this technology has received considerable attention in education for its ability to promote deep learning and collaboration between students [19]. The advantage brought by this medium is that it frees interlocutors from the limitations of time and space, thereby providing logistic flexibility for interaction and learning. However, students participate in forums by posting messages, as such, one disadvantage of the asynchronous nature of forums is that the discussion are more prone to get off-topic due to the inherent time-lags in the discussion [14]. Another problem is that manual monitoring and assessment of message contributions is time-consuming and often tedious [18]. For this reason, automated analysis for discussion understanding to enable better information assessment and assistance are much sought for [2].

In this context, the aim of our study is to develop tools that will provide assistance to teachers in detecting when conversations start to drift away from its intended topic focus. This sort of application requires an alternative method of text analysis, one which can detect the topical values of *short written headless context* [13] by relying only on small and un-annotated training data. To develop this method, we exploit lexical distributional properties together with some observable surface level text characteristics. For analysis, we use vector representation generated through Random Indexing and metrics adopted from the Gloss overlap measure proposed by Lesk [10].

This paper is organized as follows. Section 2 presents our proposed corpus-based topic model and discuss how to train this model. Section 3 touches on a relative topic alignment detection approach. Section 4 presents an initial experiment we conducted aimed at initially evaluating our proposed methodology. Finally, section 5 gives short remarks.

2. A DISCOURSE TOPIC MODEL

According to Brown and Yule [3], two basic levels of topicality exist: one is sentence topic and the other is discourse topic. The term sentence topic refers to that notion of topic associated with descriptions of sentence structure, i.e. identifying the subject-predicate components of sentences. However, as sentences are strung together, a notion of topic emerges distinct from the sentence-based topic; this notion is what can be referred to as the Discourse Topic. This type of topic is more concerned at representing the gist of the whole discourse rather than the subject focus of individual sentences. Our study is focused on the latter type of topic. We propose modeling discourse topic as a network of the most salient and related lexical items in the text; where each node represents a concept and the edges connecting them represent the weight of their pair wise association. In the following sections we describe how we propose to implement this model.

2.1 Surface-Level Statistics as Indicator of Topic

If language is a form of expression and description of ideas then the atomic means for such expression and description are content-bearing words that name corresponding notions and concepts [9]. Following this idea, we treat content-bearing words as important indicators of discourse topic and their surface level statistics as a measure of how strongly they represent the topic.

A common approach applied to words in order to determine its strength as indicators of topic is to measure its *occurrence frequency* in the text [17]. The assumption here is that frequently occurring words reflect the topic of the text more strongly than words that occur less frequently. The formula for the computation of the term frequency of a word *w* in a text *t* is simply the number of occurrences of *w* in *t*, as such:

$$tf(w, t) = occur(w) \qquad (1)$$

However, while term frequency is informative, it is not sufficient to measure the holistic importance of a word in a text. Further improvement on this measure can still be achieved by taking in consideration the word's *density property*. The basic idea of *word density* is to measure the tendency of words to repeat within a document [6]. This can be achieved by dividing the frequency of a word by the length (i.e., the total number of words) of the text, as such:

$$\textit{density}\ (w, t) = \frac{\textit{occur}(w)}{\textit{length}(t)} \quad (2)$$

It is also possible to exploit other kinds of text property such as *word burstiness* [9]. In contrast to density, burstiness focus on the distributions of distances between successive occurrences of the same word, the utility behind this measure is based in the intuition that words which express the main concept or topic of a text are used in more uniform ways than other regular words. Thus, these set of words are characterized by multiple and then often bursty occurrence. To measure this property, we follow Altman's [1] view of words in a text as being enumerated in order of appearance, $i=\{1,2...,N\}$, where i plays the role of the temporal disposition along the text. We treat i as an ordinal number assigned to each word in the text which is kept continuous across sentences and paragraphs. Given this, we can define the distance between two successive uses of a word w as:

$$ds(w_j, i) = i_{j+1} - i_j \quad (3)$$

Where $ds(w_j,i)$ is the distance score, *w* is the word in question, and *j* is the occurrence position of *w* in *i*. In this definition, the distance score is represented by the number of word positions that separate the two consecutive occurrences within the text. Our intent is to generate an estimate of the burstiness of a word by considering all the distances between all the occurrences of *w*. For this, we define a function *bs(w,t)* to determine the strength of the burstiness of a word. This is defined as:

$$bs(w,t) = \frac{1}{\sum \log(|w|)} \quad (4)$$

Where *bs(w,t)* is the burstiness score, *w* is the word in question, *t* is the text where *w* occurred, and |*w*| is the set of distance scores between all the occurrences of *w* in *t*. In effect, the function *bs(w,t)* gives higher scores to words whose occurrences are closer to each other within the text. For instance, given that the temporal appearances of a given word w_0 in a text are at the following positions: $i_1=22$, $i_2=41$, $i_3=44$, $i_4=50$. This will result to the following set of distance scores: $|w_0| = \{19, 3, 6\}$. As such, w_0 will be assigned a *burstiness score* of 4.16. If the distance distribution of another word w_1 is more dispersed, say $|w_1| = \{25, 10, 15\}$, then a lower score of 2.56 will be assigned to w_1.

Finally, to derive a topic *relevance score* which describes the overall topic bearing capability for any word in a text, we attempt to conflate the scores generated by the density and burstiness functions. This process of combining evidence is often referred to in the literature as *data fusion* [4]. For this purpose, we use the following equation.

$$rs\ (w,t) = \textit{density}\ (w, t) \times bs(w,t) \quad (5)$$

In effect, this equation ranks the relevance of words not only by looking at their frequency in proportion to the size of the text but also by considering the semantics of words by looking at their burstiness. Our use of the former factor is supported by the fact that word frequency relative to the length of the document is a good predictive measure of the ability of words to discriminate topic category; use of the latter factor, on the other hand, is primarily motivated by the assumption that the burstiness of words is directly driven by their semantics [9].

2.2 Word Space Models and Random Indexing

The word-space model is a computational model of word meaning that utilizes distributional patterns to represent semantic similarity between words [15]. The idea behind this model is that the meaning of words can be captured using spatial representation and the similarity between two words can be determined based solely on their usage in the text; this eliminates the need for any lexical annotation and enables processing in any language.

2.2.1 Random Indexing (RI)

Random Indexing (RI) is one of the methods that can be used to generate a word space model from a given corpus of text. There are two basic steps involved in using Random Indexing to implement a word-space:

1. The first step involves the assignment of a unique and randomly generated label called an index vector to each word in the data. These index vectors resemble the context vectors ordinarily generated in a word space, the only difference is that the values of the elements in these vectors are ternary. This means that they only consist of a small number of randomly distributed +1s and -1s, with the rest of the elements set to 0.
2. The second step involves generating the actual context vector that will represent each word in the vocabulary. These context vectors have the same dimensionality as the index vectors. They are automatically generated by scanning through the text, and each time a word occurs in a context, that word's index vector is added to the context vector for the focus word in question.

Random Indexing is convenient for several reasons. First, by using randomly generated index vectors, it is able to inherently control the dimension of the vector space. Second, it handles gracefully the introduction of new vocabulary; since the values of the index vectors do not change with the introduction of new texts, there is no need to recompile every time new data is inserted into the Word Space. Third, it can be implemented using very minimal resources. Finally, since it only involves simple computations, it is less expensive to use than other word space modeling techniques [15].

Random Indexing can provide a means of representing the context usage of words as derived from the textual environment. In this sense, we represent *context* as a specific text window in which the word appears. The size of the text window is equivalent to *2n+1* words, each word is denoted by W_i, where $-n<i<+n$ and W_0 is designated as the focus word.

	planetary	charged	planets	solarsystem	rings	outer	dust	satellites	stream	titan
particles	0.75	0.68	0.68	0.67	0.67	0.66	0.64	0.63	0.63	0.63

	giant	cloud	collapse	jupiter	solarsystem	nebula	saturn	planets	rock	silicates
molecular	0.792	0.783	0.760	0.754	0.751	0.749	0.741	0.740	0.717	0.705

Figure 1: Signature words of "particles" and "molecular" with their corresponding distributional similarity with the focus word, where N=10.

Thus, the context from which we approximate word usage can be viewed as an *n x n* structure where *n* represents the number of words that appear in specific positions to the right and left of a particular focus word within the textual environment. Using Random Indexing, we are able to capture this relevant linguistic information, as covered by the size of our context window, and encode them as context vectors. The context vectors in turn enable us to compare the semantic similarity between words. In this study, we assume that the degree of semantic similarity between words is also reflective of their topical alignment.

2.3 Combining the Functions of Surface Level Statistics and Random Indexing

Surface level statistics and Random Indexing can both be used to derive semantic representations that can guide linguistic topical processing. The strength of surface level statistics is that it provides a measure of how important each word is relative to the topic of a specific text. On the other hand, Random Indexing can provide a measure of how semantically related two words are based on their context usage. Our position is that the advantages of these two approaches can be combined to generate a lexical network representative of the discourse topic of the text. To this end, we borrow from Norvig's [12] basic methodology for organizing lexical items into a network of senses; this methodology has two steps as follows:

1. Select a particular semantic domain to be examined, along with the associated terminology (The selected domain must then be analyzed and described, i.e., a formal description between the associated terminologies in terms of logical relations, if possible, or some ad hoc description devices must be made)
2. Second step is to examine the usages of each lexical item in the semantic domain, i.e., for each word, we extract a concordance from the corpus, and then label each usage. The results of this analysis must be organized into a network of senses for each word, where each analyzed usage is classified under some sense.

2.3.1 Assigning Concordance of Words

In line with Norvig's first step, we first use surface level statistics to identify topically relevant keywords in the texts and filter out terms that had little topic identification value.

After this, we used Random Indexing to cull out and assign a concordance of words to each of those keywords. We refer to this concordance of words as the *signature words*[1].

In Figure 1 we can see examples of the signature words of the words *particles* and *molecular* which we derived from a small set of sample corpus downloaded from wikipedia[2]. As shown, the word-context vectors are composed by the *N* highest scored words based on distributional similarity as derived through Random Indexing. We believe that, along with their similarity scores, the *signature words* can serve as a gloss-like description of the concept of the focus word.

One drawback of this representation however, is that the similarity scores generated by RI between each of the *signature word* and the *focus word* can only be descriptive of the degree of distributional similarity between these words (i.e., their lexical closeness); it does not reflect the strength of the topical relevance of their relationship with respect to the topic of the text.

To address this issue, we propose using the relevance scores of each word as computed using (5) to supplement the RI generated distributional similarity scores. Let *w* be the focus word and $N_w = \{n_1, n_2 ... n_k\}$ be the ordered set of the top scoring *k* neighbours of *w* (i.e., the signature words) from the corpus with their associated distributional similarity scores $\{dss(w,n_1), dss(w,n_2), ...dss(w,n_k)\}$. We define the overall semantic value that each of the *signature word* contributes to the focus word using the equation:

$$\boldsymbol{semanticvalue(w,n_k) = dss(w,nk) \times rs(w,t)} \qquad \textbf{(6)}$$

By using equation (6), we believe that the resulting numerical representation combines the perspectives of two of the most important text signatures used in the representation of word semantics: 1) the distribution of lexical similarity [5], and 2) the statistical word regularities [7]. We present in Figure 2, a sample scoring of the word-context vectors of the words *particles* and *molecular* after equation (6) is applied. Notice how the relationship of the focus word with keywords that more strongly reflect the general theme of the text (i.e., *Solar System* and *Planets*) is highlighted in both vectors. We will refer to this type of vector as the *topical weight vector* because we believe they provide a good representation of the degree of importance of each word relative to the topic of the current universe of text.

[1] This concordance is similar in structure to the topic signature proposed by Lin [11], the main distinction being the manner by which the weights which describe the characteristics topical strength of each term are assigned.

[2] http://en.wikipedia.org/wiki/Solar_System

	planetary	charged	planets	solarsystem	rings	outer	dust	satellites	stream	titan
particles	0.00	0.00	0.41	0.24	0.00	0.02	0.00	0.02	0.00	0.00

	giant	cloud	collapse	Jupiter	Solar system	Nebula	Saturn	Planets	Rock	silicates
molecular	0.010	0.008	0.001	0.024	0.269	0.003	0.008	0.447	0.001	0.001

Figure 2: The corresponding topical weight vector of the words "*particles*" and "*molecular"*

2.3.2 Lexical Network Construction

Having obtained a suitable representation for each term, the final step for this stage is to recognize the implicit similarity between words and use this as a guide in generating the lexical network. Lexical networks constitute a knowledge representation formalism which represents text resources in a node-link structure. Nodes represent the sense of a word and links represent the relations between them. This kind of structure supports the machine-understandable processing of the sense of the words and can possibly provide support for scoring intra-word and inter-word relevance.

In general, our proposed approach takes as input a set of *T* terms represented by their *topical weight vectors* and outputs a network graph *T'* which we refer to as the LexNet. The proposed procedure for generating LexNet is as follows:

1. The term with the highest relevance score on the list will become the first node on the network. We will refer to this node as the *init node*.
2. All subsequent terms are compared with all the nodes created up to the current point in time. A threshold strategy is used to determine if the terms are similar enough to warrant linking the node of the new term to any specific node. If the terms are acceptably similar then an edge is created and the generated relatedness score is assigned as the value of that edge.
3. If the term does not satisfy the minimum similarity threshold for any node in the current network, then its node is treated as the *init node* for a new outlier graph.
4. The comparison process will continue until all terms in the input list have been processed and converted into a node.

3. MEASURING TOPIC ALIGNMENT

Two defining components of the above procedure are the *comparison strategy* used to compare nodes and the *thresholding strategy* used to determine whether new edges can be created between these nodes.

3.1 Comparison Strategy

A main concern of the comparison strategy is the measure that can be used to generate a value that characterizes the degree to which two words are related. There are many different definitions of measures that can be used to compute such value; our goal here is to find the appropriate function that can evaluate the relatedness between terms using their respective *topical weight vectors*.

Metrics based on spatial distances such as the cosine similarity score are not applicable for this purpose because this type of measure assume that the elements of the vectors being compared share a common reference point, i.e., the n-dimensional set of axis where the vectors are plotted. Without this assumption, there is no way that the angle between vectors can be measured correctly and thus, relatedness cannot be established. Unfortunately, such is the case in our *topical weight vector* where the values are skewed to reflect weight of topical importance instead of spatial position. Because of this, we need to come up with a new strategy on how to compare our terms.

A possible approach that we can adapt in our case is the Gloss overlap measure. Gloss in this sense is defined as some form of brief explanation or definition of obscure words usually provided in dictionaries and other lexical resources such as WordNet. Gloss overlap measures were first introduced by [10] for use in word sense disambiguation. The Lesk Algorithm compares the glosses of a pair of concepts and computes a score by counting the number of words that are shared between them. For example, it assigns a score of 1 to two concepts if there is only one word overlap between them.

Two fundamental premises underlie Lesk's gloss algorithm: The first is that words that appear together in a sentence can be disambiguated by assigning to them the senses that are most closely related to their neighboring words. The second is that related senses can be identified by finding overlapping words in their definitions. We noticed that the first premise closely resembles the hypotheses that underlie the random indexing approach. If we assume that the neighboring words that random indexing can generate for any given word can act as a sort of gloss-like signature, then it is most probable that the strategy of counting the number of overlaps between their neighbors can give us an estimate of the similarity of two words.

To our knowledge, this strategy represents the first attempt to define a measure of topical relatedness between two concepts based on the random indexing approach and using a gloss-similarity algorithm.

The general approach for our overlap scoring mechanism can be formally defined as follows:

1. Given two topical weight vectors, the overlap words between them are first detected.
2. Then, we calculate the overlap weight for each vector by computing the sum of the squared values of the overlap words and dividing this by the sum of the squared values of all the elements. The formula is shown in equation (7).

$$OLWeight(V) = \frac{\sum_{i \in n} (S_i)^2}{\sum_{j=1}^{N} (E_j)^2} \qquad (7)$$

Where *V* is the topical weight vector whose overlap weight is to be computed, n is the set of indexes of the overlapped words in V, *N* is the number of elements in V, S_i is the semantic value of the overlap word at index *i* of *V*, and E_j is the semantic value of the element at index *j* of *V*

3. Finally, the overlap weights of the two vectors are combined to arrive at the relatedness score for the given pair of vectors as shown in equation (8).

$$RelatednessScore\ (V_1,V_2) = OLWeight(V_1)\ \text{X}\ OLWeight(V_2) \quad \textbf{(8)}$$

We deem that the relatedness score generated by the above procedure enables the network to be organized along the lines of semantic similarity where two concepts, represented by their corresponding nodes are rated to be similar to each other so long as they have properties (i.e., in this case, words) in common. The more properties they share, the more closely related they are rated.

3.2 Thresholding Strategy

In tandem with the comparison strategy we also implement a thresholding strategy which influences the decision of whether or not to create a new edge between term nodes. By default, a user defined similarity threshold is used to determine whether the relatedness score generated by two words is sufficient enough to warrant linking them in the network. However, this threshold is adjusted every time a node gets new connections.

The process of adjustment is done by computing the sum of all the edge weights plus the default threshold value and dividing this by the number of edges of the node plus one. We refer to this new threshold value as the *centroid*. Whenever a new candidate term is considered for linking, the system determines whether the similarity score it generates exceeds this centroid value, if it is then the node for the new word is connected. This process is applied for all the nodes except the *init nodes*.

This thresholding strategy plays an important role not only in limiting the number of connections between nodes but also in maintaining the quality of relations encoded in the network. In particular, as the values of the edges of the nodes increases, the threshold for being accepted to that node also increases, this is expected to make the node stricter at establishing connections.

3.3 Generating Text Topical Alignment Scores

In order to automatically measure the topical alignment of concepts in an input text to that of the discourse topic represented by the LexNet, the system also identifies content-bearing keywords in the input text. The complete heuristic works in three steps, namely: (1) candidate keyword extraction, (2) computation of the relative similarity of each keyword to nodes in the network, and (3) averaging of the cumulative scores gathered by the keywords.

The candidate keyword extraction step parses the input text and extracts all the content-bearing terms. *Signature Vectors* and *Topical Weight Vectors* for the extracted keywords are generated using the same steps as in the lexical network generation.

Next, a simple word sense disambiguation is applied to the keyword by computing the similarity of its context usage to those of the network nodes: If a node for the keyword is already present in the network, then its context usage is compared to the context of the neighboring nodes of that node. Otherwise, the most likely representative node is selected by identifying the node with the most overlap words. The latter option enables the system to utilize words not found in the input text in measuring its topical alignment. If the system cannot find a node with overlapping signature words with the keyword then it simply assumes that the keyword is a non-content bearing word and assigns it a null score value (score = 0).

Finally, the overall score that indicates the relative alignment of the input text to the topic represented by the lexical network is obtained by summing up the individual scores of each word and dividing this by the number of content words processed.

We aim to apply this topical alignment scoring scheme to each message in the discourse transcript in order to generate corresponding visualizations of the topical progression of the interaction. In this sense, we are proposing that the topical progression can be visualized by measuring how relevant each message is to a common reference point represented by an external reference document.

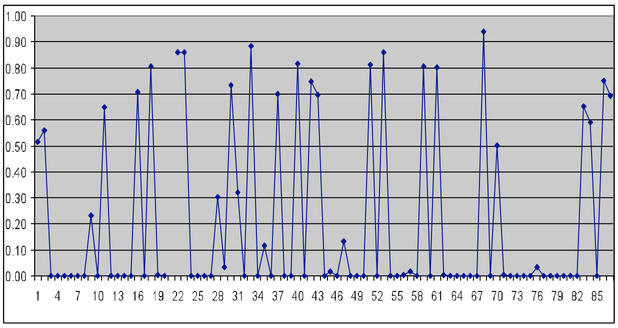


Figure 3 : Sample visualization of discourse interaction

Figure 3 shows a sample visualization generated using the highest kappa rated output of LexNet in our experiment. Sustained low points of the interaction are clearly visible in this chart. Manual verification indicates that most of these low points correspond to the phases where interlocutors are flaming each other in an off-topic mode of interaction. In academic settings, such information can be used by teachers to identify the appropriate moments to intervene and mediate the discussion.

4. EXPERIMENTS AND RESULTS

To initially test the operation of LexNet, we used a discussion transcript culled from a public forum as dataset. The discussion transcript contained a total of 87 messages with an average message length of 86.3 words. The relevance judgment for each message in the transcript was independently performed by three experienced professionals. These people were asked to rate the topical alignment of the content of each message in the discussion transcript to that of a given reference document. They were instructed to read first the reference document, after which they were asked to give a binary judgment as to whether they think each message is relevant or irrelevant to the topic expressed in the document {1 = relevant, 0 = irrelevant}. They performed this task independently, without convening with each other. Given this, the three annotators generated substantial prediction reliability with an average Kappa of 0.63. To generate a dataset with an over-all consensus rating, the decision of the third coder was used to reach a unanimous tag.

The manually rated messages served as the gold standard in our experiment. Using this, we initially assessed LexNet's performance at the message level. Cohen's Kappa was also used to determine the level of agreement of the systems rating with

Table 1: Details of the training documents used and results of the various system runs.

Text topic	SQL DataSource Control								Solar system	Spring Framework
System Run	First Set				Second Set				Third Set	
Document#	1	2	3	4	1+2	1+3	2+3	1+2+3	5	6
Size	600	1137	2707	5063	600 +1337	600+ 2707	1337+ 2707	600+ 1337+2707	6404	2517
Kappa	0.18	0.3	0.31	0.36	0.25	0.43	0.35	0.38	0.16	0.03
precision	0.57	0.53	0.53	0.63	0.54	0.64	0.56	0.61	0.53	0.38
recall	0.27	0.57	0.60	0.50	0.43	0.60	0.60	0.57	0.27	0.20

those in the gold standard; precision and recall values were also computed. We hope that this experiment would serve the purpose of identifying weak points in the function of our proposed algorithm. To achieve our objective, we treated any message with a computed topical alignment score between 0.1 and 1.0 to be relevant (category = 1). Otherwise, it was treated as not-relevant (category = 0).

There are three main parameters that can be set during a system run. First, the set of values that define the operation of Random Indexing can include: (1) the dimension of the context vectors used; (2) The degree of randomness for the index vectors; and (3) the number of nearest neighbors to consider for the signature words (i.e., the value *N*). Second, the size of the window used to identify the context for each word in the textual environment also needs to be set. Finally, the similarity threshold value for the nodes in the network also needs to be identified.

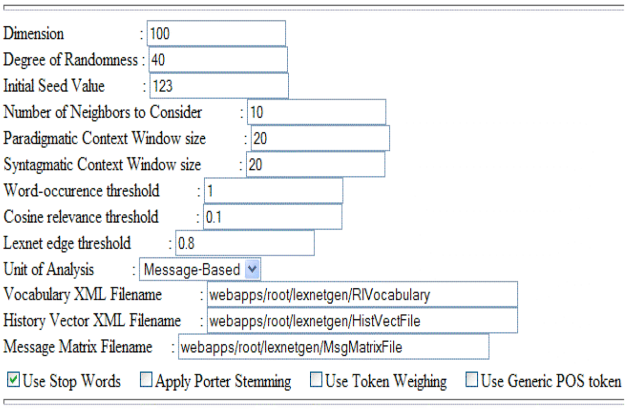
In this experiment, the parameter values we used for the Random Indexing operations are as shown in figure 4. We also set the context window size to *n* = 20 and the edge similarity threshold for LexNet to 0.8.

We conducted three sets of system runs: In the first set, we used the original text used by the human coders as reference document as training data for LexNet along with other texts, of gradually increasing sizes, that we evaluated as focusing on the same topic. This run will enable us to observe the baseline performance of the system relative to the size of the training data used.

In the second set, we tested the system on combinations of the training documents used in the first test. We wanted to determine how the simple merging of training documents will affect the system's performance. Finally, in the third set, we run the system on training documents whose topic is different from the focus of the discussion. In this case, we simply wanted to see how the system's performance will react given an irrelevant set of training documents in order to compare it to the results of the first set.

All training documents were preprocessed by removing a small set of stop words, all numbers and ordinal items, as well as other items such as web URLs. Table 1 presents details of the training documents used as well as the results of the various system runs.

Document #1 in table 1 is the original document used as basis by the human raters; document #2 - #4, however, all focus on the same topic and shares many similar keywords with #1. Between document #5 and document #6, the latter is much closer to the topic focus of the discourse transcript because it also focuses on software applications. Document #6 shares a lot of similar terms with #1 - #4, however, these terms are anchored on the java programming language while terms in #1 - #4 are anchored on ASP.NET.



Dimension : 100
Degree of Randomness : 40
Initial Seed Value : 123
Number of Neighbors to Consider : 10
Paradigmatic Context Window size : 20
Syntagmatic Context Window size : 20
Word-occurence threshold : 1
Cosine relevance threshold : 0.1
Lexnet edge threshold : 0.8
Unit of Analysis : Message-Based
Vocabulary XML Filename : webapps/root/lexnetgen/RIVocabulary
History Vector XML Filename : webapps/root/lexnetgen/HistVectFile
Message Matrix Filename : webapps/root/lexnetgen/MsgMatrixFile
☑ Use Stop Words ☐ Apply Porter Stemming ☐ Use Token Weighing ☐ Use Generic POS token

Figure 4: Parameter values used in the initial system run

So far, looking at the Kappa results of the first set of system run under document #1 - #4 and comparing it to the results of the third set under document #5 and #6, we can initially claim that LexNet's ability to distinguish the topical relevance of forum messages is directly linked to the topic focus of the contents of the training document. Also, the performance of the system under the first set clearly shows improvement as the size of the training document is increased. The next question to address then is how to effectively increase the size of the training document. The most obvious way is to simply feed the system with multiple documents that focuses on the same theme.

However, Kappa results generated in the second set seem to indicate that this approach is not as straightforward as it seems. Although the highest Kappa value in this experiment was generated by combining documents (#1 and #3), the same process also degraded the performance of the system in other examples, e.g., 1+2 as compared to 1 and 2; and 1+2+3 as compared to 1+3. If our assumptions are correct that the LexNet is structured along the lines of semantic similarity through common properties, then, we can probably postulate an explanation for this performance degradation through the conjecture that writers only use one sense of a word throughout a text. Such senses are implicitly expressed by the writer through the unique combinations of content keywords used. This conjecture conforms to the recent findings of Gale [8].

Given this, we can further posit that combining the contents of two texts whose keyword senses are well-matched will result in an improved system performance. On the other hand, merging texts whose content keyword combinations do not match will result to a degraded system performance because the new text will only introduce new senses which will make words in the other text more ambiguous to the system. This observation, we believe, also conforms to Brown and Yule's suggestion that not all available background knowledge needs to be used in analyzing a fragment of discourse. Only those which are directly related to the current discourse in consideration needs to be accessed.

Overall, results of the experiment indicate that the output of LexNet is still not at par with the decisions of the human coders. This output can still be improved however, by increasing the size of the training data; although careful filtering of the combinations of documents is clearly needed. Another means by which improvement can be achieved is to exploit other resources that can be used to decipher the topical value of message contributions. We believe that one reason why the human coders generated an acceptable average Kappa value is because they were able to use the background knowledge generated by the previous messages. Brown and Yule referred to this linguistic resource as the *domain of discourse*. According to them [3]:

> "*the initial setting of the co-text (the domain of discourse) determines the extent of the context within which the hearer will understand what is said next*".

Our system is not yet able to take advantage of this resource, however, for future work we intend to integrate this into the topical alignment scoring mechanism. We believe that this will induce further improvement on the performance of the system.

5. CONCLUDING REMARKS

In this paper, we presented our idea of how surface-level statistics of text characteristics can be combined with the context usage of words to generate a lexical network that can be used to measure the intra- and inter-sense relationships of words. Based on this, we obtained a simple approach to assessing the relative degree of topical alignment of two texts and applied it to visualizing the topical alignment between messages in a discourse transcript.

Although the results generated are rather low, still, we believe that it is indicative of the potential of the proposed approach. However, at this point, the identifiability of the model is still an issue. Many more parameters need to be considered and tested and several open problems remain. Further experimental studies are clearly needed.

6. REFERENCES

[1] Altmann, E.G., Pierrehumbert, J.B. & Motter, A.E. 2009 Beyond Word Frequency: Bursts, Lulls, and Scaling in the Temporal Distributions of Words. *PLoS ONE* 4(11): e7678. doi:10.1371/journal.pone.

[2] Barros, B. & Verdejo, F. 2000. Analysing Student Interaction Processes In Order To Improve Collaboration. The DEGREE Approach". *International Journal of Artificial Intelligence in Education*.

[3] Brown, G. and Yule, G. 1983. *Discourse Analysis*. Cambridge Textbooks in Linguistics Series. Cambridge University Press.

[4] Croft, W. B. 2000. Combining Approaches to Information Retrieval. In W. B. Croft (Ed.), *Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval*. Kluwer Academic Publishers.

[5] Cruse, D.A. 1986. *Lexical Semantics*. Cambridge University Press, 1986. ISBN 0-521-27643-8

[6] Franz, M. & McCarley, J.S. 2000. Word Document Density and Relevance Scoring. In *SIGIR '00: Proceedings of the 23rd Annual International ACM SIGIR Conference*, 345–347, Athens, Greece, July 24-28, 2000. ACM Press.

[7] Furnas, G.W., Landauer, T.K., Gomez, L.M. & Dumais, S.T. 1983. Statistical Semantics: Analysis of the Potential Performance of Keyword Information Systems. *Bell System Technical Journal*, 62(6):1753-1806.

[8] Gale, W.A., Church, K.W. & Yarowsky, D. 1992. One Sense per Discourse. In *Proceedings of the 16th International Joint Conference* (IJCAI), 233-237, Los Altos, California, 1992. Morgan Kaufmann.

[9] Katz, S.M. 1996. Distribution of Content Words and Phrases in Text and Language Modeling. *Natural Language Engineering*, 2(1):15–59. Cambridge University Press

[10] Lesk,M. 1986. Automatic Sense Disambiguation using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, 24 - 26.

[11] Lin, C.Y. 1997. Robust Automated Topic Identification. *PhD Computer Engineering Dissertation*. University of Southern California.

[12] Norvig, P. 1989. Building a Large Lexicon with Lexical Network Theory. In *Proceedings of the IJCAI Workshop on Lexical Acquisition*, August 1989.

[13] Pedersen, T. 2008. Computational Approaches to Measuring the Similarity of Short Contexts: A Review of Applications and Methods. To appear in the *South Asian Language Review*, CoRR abs/0806.3787:(2008).

[14] Potter, A. 2007. An Investigation of Interactional Coherence in Asynchronous Learning Environments. *PhD Dissertation*, School of Computer and Information Sciences, Nova Southeastern University.

[15] Sahlgren, M. 2006. The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-dimensional Vector Spaces. *Ph.D. thesis*, Stockholm University.

[16] Sahlgren, M. 2005. An Introduction to Random Indexing. In *Methods and Applications of Semantic Indexing Workshop, 7th International Conference on Terminology and Knowledge Engineering*.

[17] Salton, G. and Buckley, C. 1988. Term-weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5):513-523.

[18] Wu, Y.B. & Chen, X. 2005. Assessing Student Learning With Automated Text Processing Techniques. *Journal of Asynchronous Learning Networks*, 9(3), October 2005.

[19] Wu, D. & Hiltz, S.R. 2004. Predicting Learning from Asynchronous Online Discussions. *Journal of Asynchronous Learning Networks*, 8(2), April 2004.