Detecting Hidden Passages in Documents

Saket S.R. Mengle
Information Retrieval Laboratory
Computer Science Department
Illinois Institute of Technology
Chicago, Illinois, U.S.A
saket@ir.iit.edu

Nazli Goharian
Information Retrieval Laboratory
Computer Science Department
Illinois Institute of Technology
Chicago, Illinois, U.S.A
nazli@ir.iit.edu

ABSTRACT

Passages can be hidden within a text to circumvent their disallowed transfer. Such release of compartmentalized information is of concern to all corporate and governmental organization. We present our methodology to detect such hidden passages within a document. A document is divided into passages using various document splitting techniques, and a text classifier is used to classify such passages. Our detection rate, as shown empirically, is 76% with an equivalent precision. We provide a comparison of various passage identification methods and also evaluate the effects of passage length and feature selection in this process.

1. INTRODUCTION

Transferring information outside organizational boundaries is a concern to both commercial and governmental organizations [2]. Such information can be hidden as passages within a text. It is not feasible to manually check for such passages within large documents.

Traditionally, text classifiers are used to identify the topic of a document. Text classifiers treat each document as a single classification unit and assign one or more categories to that document. However, a document may contain hidden passages whose contents differ from the assigned category of that document. Though text classifiers work effectively to assign categories to documents, they fail to identify such hidden passages.

We use a three-phase methodology for hidden passage detection. In the first phase, training documents are used to build a text classification model based on the document terms and a priori known categories of these documents. In the second phase, we preprocess the documents by dividing a document into passages using the well-known document splitting techniques (*window passage approach*, *overlapping window passage approach* and *discourse passage approach*). In the third phase, the text classification model is used to detect the infected documents, i.e., the documents that contain a passage related to a user specified category. User specified category is defined as a category, related to which, the user is interested to detect hidden passages. In the context of this work, user specified categories are related to malicious topics such as topics on terrorism, war, computer hacking, etc. Details of methodology are provided in Section 3.

Passage retrieval research efforts [1][5] have addressed approaches to find passages in a document that matched a user query, or even expanded user query such as using relevance feedback. However, the passage retrieval approaches do not identify the passages based on the subject matter, or category of content of such passages. Our focus is on passage detection and not passage retrieval, and thus, we provide a differentiation of the two:

- Passage detection attempts to identify passages related to user specified topics (category), while passage retrieval concerns with passages related to user queries.

- In passage detection, training documents are used to train a classifier on a topic, while passage retrieval is generally not a supervised process.

- In passage detection, the effectiveness of results depends on the accuracy of the text classification model. In passage retrieval, the effectiveness of results depends not only on the engine but also on how the query is formulated by a user.

2. PRIOR WORK

A model for passage based text classification was proposed in [7] that categorizes a document based on the category of the majority of passages in that document. The objective of the work presented in [7] is to classify a document as a whole. The objective of our work is to identify the category of each passage in a document, regardless of the category of the whole document, to detect hidden passages inserted by a malicious user. We evaluate our approach based on how accurately we detect such passages.

As passages are located at random locations in a document, identifying the boundaries of passages is critical. Various techniques are used to split a document into passages. Some techniques assume that the boundary of a passage is predefined based on discourse information in a passage. The effort in [13] assumes that <*p*> and </*p*> HTML tags mark the start and the end of a passage, respectively. The discourse information like a sentence or group of sentences is also used to define a passage [1][4]. However, there are a few drawbacks in using the discourse information to define passages. First, there may be discourse inconsistency among authors [1]. Second, it may be impossible to create discourse passages, if the discourse information like punctuation marks or HTML tags is not provided with a document [6]. Finally, the discourse passages can be very short or very long based on the author's style.

To overcome these drawbacks, window based passage techniques are used to identify passages. The *non-overlapping window* [3] *passage* technique and *overlapping window passage* [1] technique assume that passages are not bounded by any delimiters and divide the document into passages of fixed window size. A detailed explanation about different document splitting techniques is given in Section 3.1. In our work, we assume that the passages that are hidden are not bounded by delimiters.

3. METHODOLOGY

A document is structured in a sequence of sub-topical discussions that occur in context of one or more main topic discussions [4]. Thus, we divide the document into passages based on the units such as sentence, contiguous sentences or contiguous text blocks.

Figure 1 presents a block diagram of our methodology. A document is divided into passages using a splitting algorithm. We empirically evaluate three algorithms to split a document into passages. Once a document is split into passages, each passage is individually classified using a text classifier. We are interested to find documents that contain passages that belong to *category x*. If the text classifier finds a passage with *category x*, it marks the document as *infected*, otherwise it marks as *clean*. This process is divided into three phases as shown in Figure 2, and is described in Sections 3.1, 3.2 and 3.3, respectively.

3.1 Phase I: Building a text classification model

We use the FACT (Fast Algorithm for Classifying Text) classifier [10] as the classification algorithm in our passage detection approach. FACT is a statistical text classifier that uses a feature selection algorithm called *Ambiguity Measure (AM)* [11]. Formally, Ambiguity measure (AM) is defined as the probability that a term falls into a particular

Figure 1: Block diagram of passage detection method

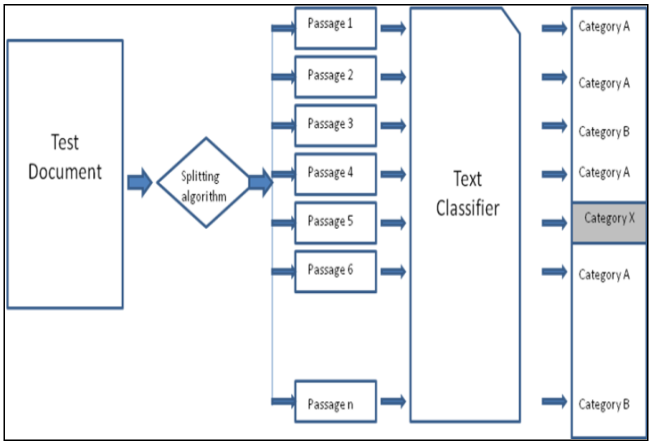


Figure 2: Algorithm for passage detection

Input:

a) User specified category (this is the category that we want to find if a malicious user has inserted a passage on this topic in a document).

b) Documents for training the text classifier containing documents that are labeled with various categories including categories that we may consider "malicious".

c) Documents that are to be tested to identify hidden Passages within them.

Output:

a) Infected documents, i.e. documents containing passages related to user specified categories.

Phase I

a) Build a text classification model using training documents on user specified (malicious) categories as well as other (non-malicious) categories. (See Section 3.1 for detailed explanation).

Phase II

a) Parse the input documents to be tested.

b) Split the document into passages using a document splitting technique (See Section 3.2 for detailed explanation).

Phase III

a) Classify each passage that is generated in phase II, using the text classification model built in phase I.

Mark the documents that contain a passage related to user specified category as *infected* and the documents that do not contain passages related to user specified category as *clean*. (See *Section* 3.3 for detailed explanation).

category and is calculated using the Formula 3.1.1 and 3.1.2. A term is considered less ambiguous if its *AM* value is closer to 1. Conversely, if its *AM* is closer to 0, the term is considered more ambiguous with respect to a given category. In the training phase the ambiguity measure of each term that occurs in training documents is calculated.

$$AM(t_k, C_i) = \left(\frac{tf(t_k, c_i)}{tf(t_k)} \right) \qquad \text{... 3.1.1}$$

$$AM(t_k) = \max(AM(t_k, C_i)) \qquad \text{... 3.1.2}$$

where, *tf* (t_k, c_i) is the number of times a term t_k appears in category c_i and $tf(t_k)$ is the number of times a term t_k appears in the entire dataset.

The detail on how the classifier is trained using our dataset is given in Section 4.

3.2 Phase II: Splitting algorithms

A passage is defined as any sequence of text from a document [6]. As the definition of passage is vague, different types of automatic document splitting techniques exist. In this work, we experiment using the following three different document splitting techniques:

1. Discourse passages
2. Window passages
3. Overlapping window passages

A detailed explanation about each document splitting technique is given in section 3.2.1, section 3.2.2 and section 3.2.3, respectively.

3.2.1 Discourse Passage Approach

Discourse passages are based on logical components such as discourse boundaries like a sentence or a paragraph [1][4]. An example of *discourse passage* approach is shown in Figure 3. In this example, a document is split into three passages such that each passage contains one sentence. In our experiments, we use different variations of *discourse passage* approach. A document is split into passages of *n* sentences where *n* = {1,2,3,4,5}.

Figure 3. Example of discourse passage where a document is divided into passages based on sentence boundaries (n=1)

Passage 1	Passage 2	Passage 3
The sky is blue.	How beautiful!	It was cloudy yesterday.

3.2.2 Window Approach

Figure 4. Example of window passage where each passage has same number of words (n=4)

Passage 1	Passage 2	Passage 3
The sky is blue.	However, it is raining	a lot since morning.

Unlike the *discourse passage* approach where passages are determined based on the structural properties of document, the *window based passage* approach defines a passage as *n* number of words. [3] proposes the window passage approach where documents are segmented into evenly sized blocks. An example of window passage approach is shown in Figure 4. There is no shared area between two adjacent windows, and hence, these windows are called *non-overlapping* windows. We experiment using different window sizes (5, 10, 15, 20, and 25 words). The effect of different window sizes on the accuracy of passage detection is presented in Section 5.1.

3.2.3 Overlapping Window Approach

The *Non-overlapping window passage* approach may break a passage that relates to a user specified category into two passages. In this case, each of the passages may contain words that do not logically belong to that passage. Thus, the classification accuracy decreases. To avoid such situations, [1] proposed the concept of *overlapping windows*. In the *overlapping window passage* approach, a document is divided into passages of evenly sized blocks by overlapping *n/2* from the prior range and *n/2* from the next range.

Figure 5. Example of the overlapping window passage where each passage has same number of words and overlap windows are also present (n=4)

Passage 1	Passage 3	Passage 5
The sky is blue.	However, it is raining	a lot since morning

	Passage 2	Passage 4	
	is blue. However, it	is raining a lot	

In Figure 5, we show an example of *overlapping window passage* approach. Similar to the *non-overlapping window passage* approach, we experiment using windows of different sizes and evaluate their effectiveness in passage detection (Section 5.1).

In our future work, we plan to evaluate other document splitting techniques to identify the method that is best suited for passage detection.

3.3 Phase III: Classifying passages

The classification model built in *phase I* is used for individually classifying each passage as was identified based on document splitting techniques, described in Section 3.2. The FACT classifier bases its decision on the most unambiguous words in a passage. Thus, even when a passage is very small, FACT classifies the passage only if unambiguous words exist in that passage. This reduces the number of false positives generated during passage detection. Moreover, as FACT uses feature selection, we evaluate the effect of feature selection on the effectiveness of passage detection (Section 5.2).

4. EXPERIMENTAL SETUP

In this section, we discuss the experimental framework used for evaluating our approaches to detect passages that belong to a user specified category. We provide the information about the training and testing dataset that is used to build the classification model and detect passages in Section 4.1. Evaluation measures that are used in our experiments are explained in Section 4.2.

4.1 Dataset

To validate our passage detection accuracy, we need a dataset, where each inserted (malicious) passage within any document is tagged with a pre-defined category. To our knowledge, no such dataset is available. Hence, we modified the standard 20 Newsgroups (20NG) dataset [8] that contains news articles about various topics like sports, electronics, science, etc for our task. Passages extracted from security related news articles on *www.cnn.com* are used to insert into some documents (test documents) in the 20NG dataset. We call these documents as *infected documents*. Documents from the 20NG dataset and Security dataset are used to train a text classifier (Section 4.1.1). To ensure better performance of a text classifier, only the non-infected documents are used for training. In the testing phase, we use 1,000 infected documents and 1,000 non-infected documents. Hence, we use a 9-1 split for the 20NG dataset instead of 10-fold cross validation so that only the non-infected documents are used for training. The statistics about the datasets that are used in our experiments are given in Table 1. Section 4.1.1 provides more details about the training documents. Section 4.1.2 provides more details on the testing documents that are generated after inserting passages on “security” topics in the 20 Newsgroups documents.

4.1.1 Training Documents

Two datasets are used for training the text classifier. We used 20 NG dataset to train the text classifier to be able to detect passages that are related to categories present in 20 NG dataset. Moreover, to train the text classifier on topics related to “security”, we created a dataset that contains documents related to *security* topics. In our system, we are concerned to detect passages related to security topics, if they are inserted by a malicious user within a document. Details about both these datasets are given below.

20 Newsgroups

20NG dataset [8] consists of a total of 20,000 documents that are categorized into twenty different news groups. Each category contains 1,000 documents. We use a random stratified 9-1 train-test split such that 18,000 documents are used for training a text classifier and 2,000 documents are used for testing.

Security Dataset

We created a dataset related to “security” topics to train the text classifier to be able to detect such topics. We created a text corpus of 3067 news articles on “security” from *www.cnn.com* containing 6 categories. As shown in [9], removing noisy text in the navigational bar improves accuracy; similarly, we removed such text and used only the news story available on the webpage. The details about this dataset are given in Table 1.

Two human evaluators assessed all 3067 security news articles and analyzed documents as *relevant, not relevant* or *undecided* to each of the 6 categories. Before doing the evaluations, the evaluators agreed upon the definition of each category. The average Pearson’s co-relation between the evaluations of both the human evaluators was 90.60%.

4.1.2 Testing documents

To simulate an environment where the administrator is interested to detect infected documents, we inserted passages into 1,000 documents belonging to 20NG dataset that were used for testing. In these preliminary experiments, we infect the documents with at most one passage. To observe the effects of our algorithm on passages of different length, we inserted passages of

Table 1: Security data set characteristics

Category	Number of documents	Description
Computer Crimes	329	About computer crimes like hacking and viruses.
Terrorism	920	About terrorist attacks and counter measures to prevent terrorism
Drugs Crimes	601	About drug trafficking and crimes related to drugs.
Pornography	344	About issues related to pornography
War Reports	342	Reports on various wars going on around the world
Nuclear Weapons	531	Reports about nuclear programs of various countries.

Table 2. Statistics about datasets

Purpose	Dataset	Number of documents	Is the document infected?	Length of passage
Training	20 NG	18,000	-	-
	Security Dataset	3067	-	-
Testing	20 NG	1000	No	-
	20 NG	200	Yes	10 words
	20 NG	200	Yes	20 words
	20 NG	200	Yes	30 words
	20 NG	200	Yes	40 words
	20 NG	200	Yes	50 words

various sizes (10 words, 20 words, 30 words, 40 words and 50 words) in the original 20NG documents. Every passage is inserted at a random word boundary location in a document. Hence, it is hard to detect the boundaries of that passage. Discourse boundaries like HTML tags are filtered out of the passages that are inserted. Each passage that is inserted is evaluated by two graduate students to verify if it indeed relates to "security" information. Table 2 shows statistics of the 2000 testing documents from the modified 20NG dataset with respect to the presence of a passage related to "security" topic and length of such passages.

4.2 Evaluation measures

To evaluate the effectiveness of our approach, we use the commonly used evaluation metrics: precision, recall and F1 measure. Precision is defined as how accurately a system predicts whether a document contains a passage related to user specified category (Formula 4.1). Recall is defined as the ratio of number of correctly predicted documents that have hidden passages to the total number of documents that have passages (Formula 4.2). F1 measure is a common measure in text classification that combines recall and precision into a single score with an equal importance (Formula 4.3)

$$\text{Precision(P)} = \frac{\mathit{True\ Positive}}{\mathit{True\ Positive} + \mathit{False\ Positive}} \quad ..\ 4.1$$

$$\text{Recall(R)} = \frac{\mathit{True\ Positive}}{\mathit{True\ Positive} + \mathit{False\ Negative}} \quad ..\ 4.2$$

$$\text{F1measure} = \frac{2PR}{P+R} \quad ..\ 4.3$$

Table 3: Contingency matrix for passage detection

			Predicted		
			Infected		Not Infected
			Passage with category x	Passage with category $\bar{x}$	
Actual	Infected	Passage with category x	TP	TP	FP
		Passage with category $\bar{x}$	TP	TP	
	Not Infected		FN		TN

Table 4: Contingency matrix for passage category prediction

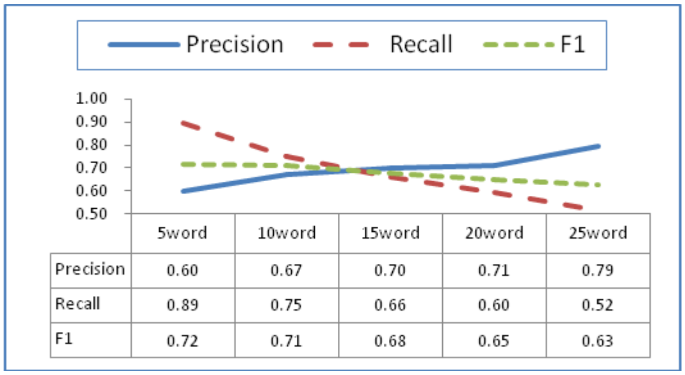
			Predicted		
			Infected		Not Infected
			Passage with category x	Passage with category $\bar{x}$	
Actual	Infected	Passage with category x	TP	FN	FP
		Passage with category $\bar{x}$	FP	TN	
	Not Infected		FN		TN

We evaluate our algorithms using two scenarios. In the first scenario, we consider true positive for an instance where a document contains an infected passage and the classifier marks the document as *infected*. We call this task as *passage detection*. The contingency matrix for passage detection is shown in Table 3. For example, if a document contains a passage related to *war* and the classifier marks the passage as *infected*, this instance is considered as true positive. In the second scenario, we consider true positives for an instance only when the classifier correctly predicts the category of the hidden passage in infected document. We call this evaluation method as *passage category prediction*. The contingency matrix for passage category prediction is shown in Table 4. For example, an instance is considered as true positive only if a document contains a hidden passage related to *war* and the classifier correctly classifiers that passage as *war* category.

5. RESULTS AND ANALYSIS

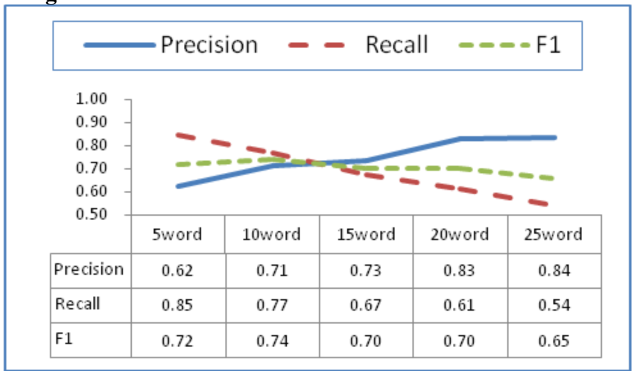
We provide our results and analysis of three document splitting techniques on the accuracy of *passage detection* and *passage category prediction* in Section 5.1. We also

Figure 6: Passage Detection results for window approach



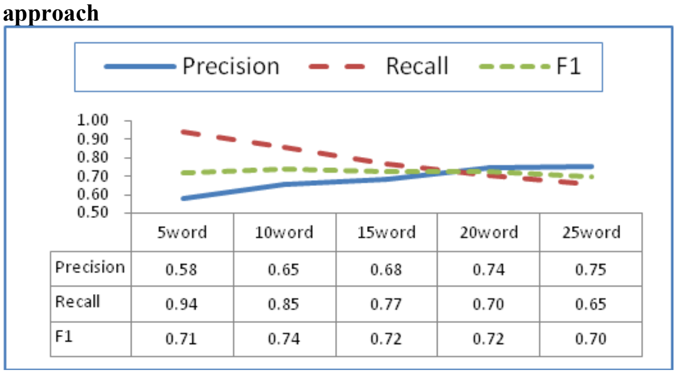
	5word	10word	15word	20word	25word
Precision	0.60	0.67	0.70	0.71	0.79
Recall	0.89	0.75	0.66	0.60	0.52
F1	0.72	0.71	0.68	0.65	0.63

Figure 7: Passage Detection results for window approach using feature selection



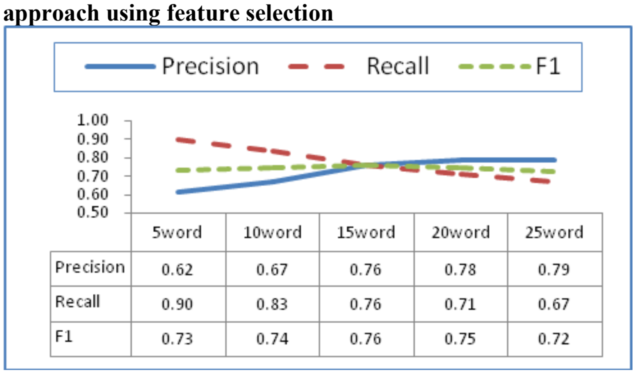
	5word	10word	15word	20word	25word
Precision	0.62	0.71	0.73	0.83	0.84
Recall	0.85	0.77	0.67	0.61	0.54
F1	0.72	0.74	0.70	0.70	0.65

Figure 8: Passage Detection results for overlapping window approach



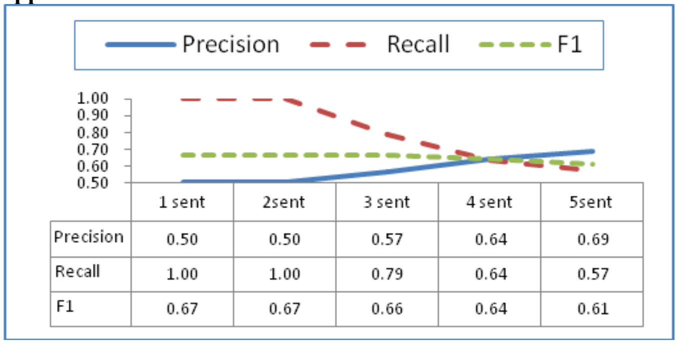
	5word	10word	15word	20word	25word
Precision	0.58	0.65	0.68	0.74	0.75
Recall	0.94	0.85	0.77	0.70	0.65
F1	0.71	0.74	0.72	0.72	0.70

Figure 9: Passage Detection results for overlapping window approach using feature selection



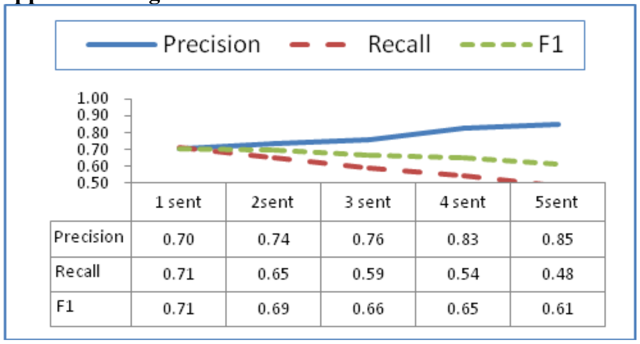
	5word	10word	15word	20word	25word
Precision	0.62	0.67	0.76	0.78	0.79
Recall	0.90	0.83	0.76	0.71	0.67
F1	0.73	0.74	0.76	0.75	0.72

Figure 10: Passage Detection results for discourse passage approach



	1 sent	2sent	3 sent	4 sent	5sent
Precision	0.50	0.50	0.57	0.64	0.69
Recall	1.00	1.00	0.79	0.64	0.57
F1	0.67	0.67	0.66	0.64	0.61

Figure 11: Passage Detection results for discourse passage approach using feature selection



	1 sent	2sent	3 sent	4 sent	5sent
Precision	0.70	0.74	0.76	0.83	0.85
Recall	0.71	0.65	0.59	0.54	0.48
F1	0.71	0.69	0.66	0.65	0.61

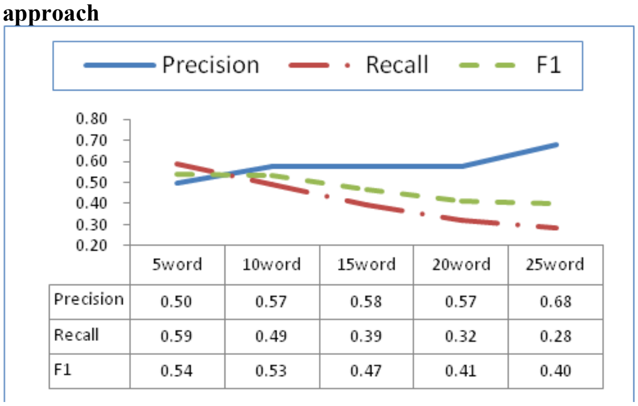
evaluate the effect of feature selection in each case. In Section 5.2, we demonstrate the effect of passage length on the detection rate. In section 5.3, we present the effect of our training model on passage category prediction.

5.1 Results of document splitting techniques

In this section, the effects of different document splitting approaches on the effectiveness of *passage detection* and *passage category prediction* are presented.

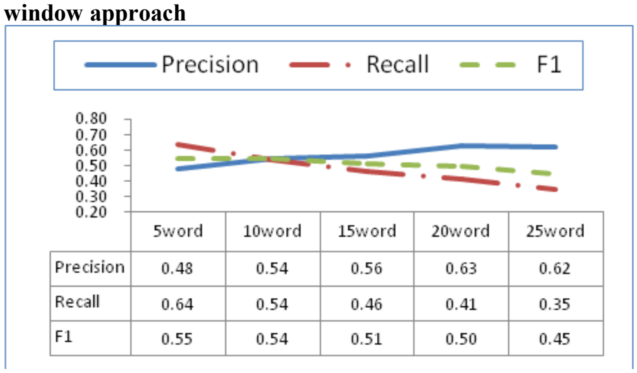
Figure 6 and Figure 7 show the results for *non-overlapping window passage* approach with and without feature selection, respectively. Feature selection is done using the ambiguity measure described in Section 3.1. The threshold of 0.6 was shown to perform the best for the dataset indicates that all the terms whose term weight (Ambiguity measure) is below 0.6 are filtered out of the feature set. The term weights are normalized between 0 – 1, where 1 indicates the highest weight of a term and 0 indicates lowest weight of a term. The X-axis indicates different window sizes (5, 10, 15, 20, and 25 word) that were used for experimentation. As the size of window increases, the precision of passage detection also increases (Figure 6 and

Figure 12: Passage category prediction results for window approach



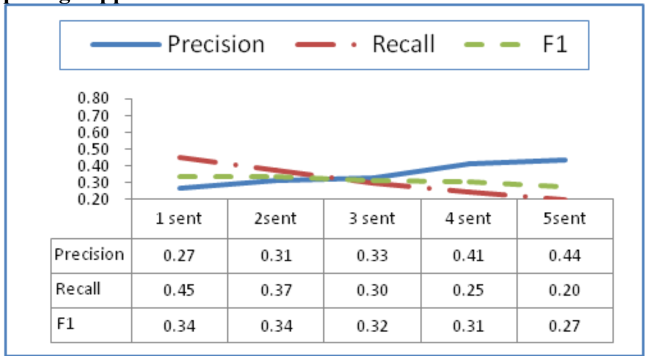
	5word	10word	15word	20word	25word
Precision	0.50	0.57	0.58	0.57	0.68
Recall	0.59	0.49	0.39	0.32	0.28
F1	0.54	0.53	0.47	0.41	0.40

Figure 14: Passage category prediction results for overlapping window approach



	5word	10word	15word	20word	25word
Precision	0.48	0.54	0.56	0.63	0.62
Recall	0.64	0.54	0.46	0.41	0.35
F1	0.55	0.54	0.51	0.50	0.45

Figure 16: Passage category prediction results for discourse passage approach

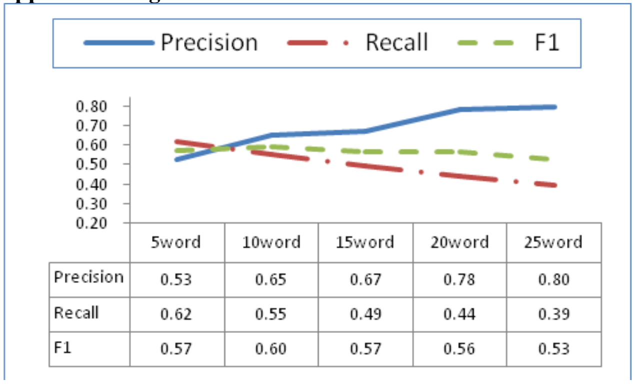


	1 sent	2sent	3 sent	4 sent	5sent
Precision	0.27	0.31	0.33	0.41	0.44
Recall	0.45	0.37	0.30	0.25	0.20
F1	0.34	0.34	0.32	0.31	0.27

Figure 7). For larger window sizes the classifier uses more words to make the classification decision. Hence, the precision of detecting passages improves. On the other hand, when the window size is large, smaller passages that are present in a document are not detected, resulting in a decrease in recall. Similar trends can be observed for *passage category prediction* as shown in Figure 12 and Figure 13.

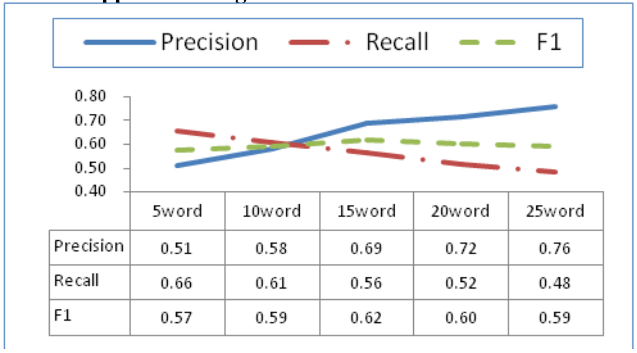
Figures 8, 9, 14 and 15 demonstrate that *overlapping window passage* follows similar trends as *non-overlapping window passage* approach for different window sizes.

Figure 13: Passage category prediction results for window approach using feature selection



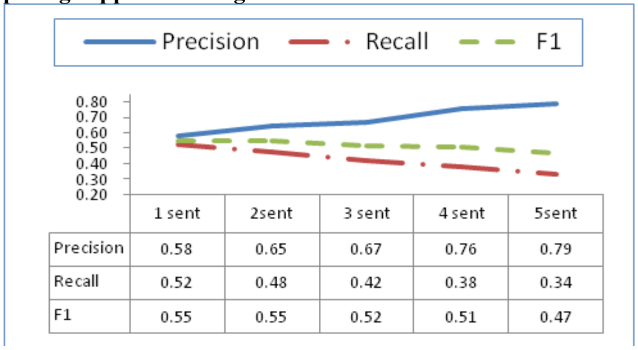
	5word	10word	15word	20word	25word
Precision	0.53	0.65	0.67	0.78	0.80
Recall	0.62	0.55	0.49	0.44	0.39
F1	0.57	0.60	0.57	0.56	0.53

Figure 15: Passage category prediction results for overlapping window approach using feature selection



	5word	10word	15word	20word	25word
Precision	0.51	0.58	0.69	0.72	0.76
Recall	0.66	0.61	0.56	0.52	0.48
F1	0.57	0.59	0.62	0.60	0.59

Figure 17: Passage category prediction results for discourse passage approach using feature selection



	1 sent	2sent	3 sent	4 sent	5sent
Precision	0.58	0.65	0.67	0.76	0.79
Recall	0.52	0.48	0.42	0.38	0.34
F1	0.55	0.55	0.52	0.51	0.47

Figures 10 and 11 show the *passage detection results* for *discourse passage* approach and figures 16 and 17 show the *passage category prediction* results for the same. In *discourse passage* approach, we group *n* sentences as one passage (where, n={1,2,3,4,5}). A sentence is defined as a unit of word sequences separated by a period. As the number of sentences in a passage increases, more words are available for a text classifier to base its decision on. Hence, the precision of passage detection increases. However, if many sentences are grouped as a single passage, the short passages in a document are ignored. Hence, the recall of passage detection decreases.

Thus, it can be observed that an increase in the size of a window in the window-based approaches or an increase in the number of sentences in *discourse passage* approach results in an improvement in precision and subsequently a decrease in recall of passage detection.

Accuracy Comparison (without feature selection)

Table 5 and Table 6 shows the comparison between effectiveness of various document splitting techniques for passage detection. As depicted in Table 5 and Table 6, the *overlapping window* approach performs statistically significantly (99% confidence) better than both *non-overlapping window passage* approach and *discourse passage* approach.

Non-overlapping window approach may have some degree of loss of information due to the fact that a passage may be split and become part of adjacent windows. The *overlapping window* approach avoids such loss of information since it also generates passages that overlap with adjacent passages. Hence, *overlapping window passage* approach performs significantly better than non-overlapping window approach.

Discourse passage approach performed statistically significantly worse than both *non-overlapping window passage* approach and *overlapping window passage* approach. As mentioned in Section 4, all the discourse information such as delimiters and passage tags were removed from the inserted passages. Hence, detecting passages that do not contain discourse information in them is difficult using *discourse passage* approach.

Feature Selection

As shown in Table 5 and Table 6, using feature selection significantly (99% confidence) improves the effectiveness of passage detection and passage category prediction with respect to precision and F1 measure. Feature selection prunes words with a lower weight from the feature set of a text classifier and only keeps the most important terms. Thus, fewer terms are available for making a decision.

However, as the decision of a classifier is based on the most important terms in a passage, a classifier only predicts a category for a passage when important terms are present in a passage. Hence, the number of false positives decreases and precision increases. However, as many of unimportant terms (terms with a low ambiguity measure value) are filtered, some of the passages that point to the "security" topics, but do not have many important terms are not detected. Hence, the recall of passage detection decreases. Nevertheless, as indicated by the results, feature selection significantly improves precision and F1 measure.

Table 5. Comparison of effectiveness of different document splitting techniques on *passage detection*

	Without Feature Selection			With Feature Selection (Results for best threshold)		
Method	**P**	**R**	**F1**	**P**	**R**	**F1**
Window	0.6019	0.8920	0.7188	0.6903	0.7800	0.7324
Overlapping window	0.6903	0.7800	0.7324	0.7613	0.7560	0.7587
Discourse	0.5000	1.0000	0.6667	0.6991	0.7110	0.7050

Table 6. Comparison of effectiveness of different document splitting techniques on *passage category prediction*

	Without Feature Selection			With Feature Selection (Results for best threshold)		
Method	**P**	**R**	**F1**	**P**	**R**	**F1**
Window	0.4992	0.5880	0.5399	0.6498	0.5510	0.5963
Overlapping window	0.4796	0.6360	0.5469	0.6853	0.5640	0.6188
Discourse	0.3108	0.4510	0.3680	0.5792	0.5230	0.5497

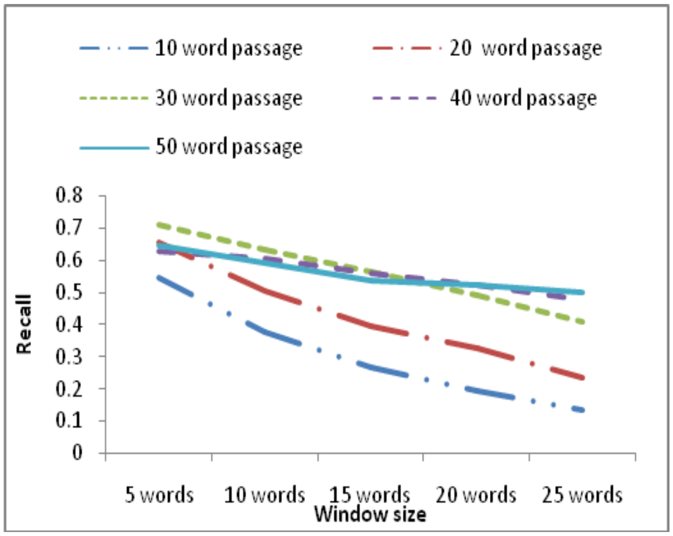
5.2 Passages of varying length

We now analyze how the size of a hidden passage in a document affects the recall of passage detection techniques. We are interested in detection rate of the passages of a given length. Hence, all the values that are discussed in this section are recall values. In our modified 20 Newsgroups dataset, passages of varying length (10, 20, 30, 40, and 50 words) were inserted into original documents.

As the *overlapping window passage* approach is our best performing method, we present only the results of *overlapping window passage*. However, similar trends are observed for both *non-overlapping window passage* and *discourse passage*.

The results of varied length passages with different window sizes in *overlapping window passage* are shown in Figure 18. The X-axis represents different window sizes and the Y-axis represents recall value for each run. As shown, for 5-word window, 30-word passage performs significantly better than other passages. However, as the size of the window increases, smaller passages are ignored and larger passages are detected better than the smaller passages. Hence, as the size of window increases, the recall for 50-word passage decreases at a lower rate than 30-word or 40 word passages. This trend indicates that the knowledge about the size of a hidden passage is important in selecting the window in *overlapping* approach. In the large passages,

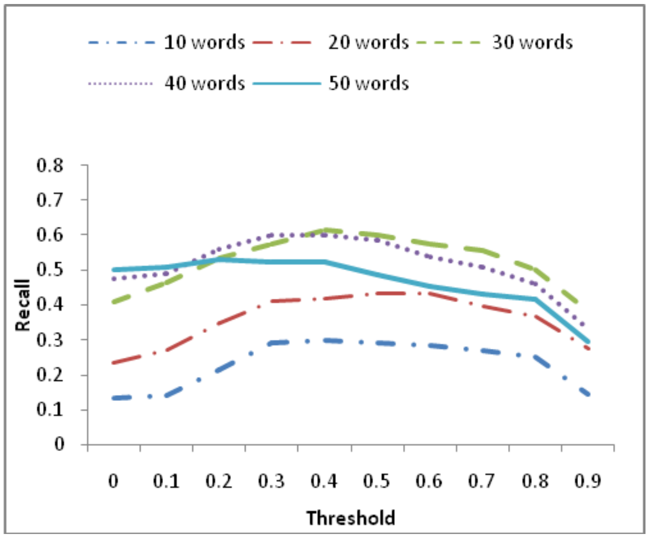
Figure 18. Recall values with respect to various lengths of passages for different window sizes in overlapping window approach



important information is sparse while information is dense in shorter passages. Hence, it is difficult to detect large passages using small window size of 10 or 20 words.

Figure 19 demonstrates the effects of feature selection on passages of different lengths while using *overlapping window* approach with a window size of 25 words. X-axis in Figure 19 represents different threshold values and Y-axis represents recall values. As observed from Figure 19, the feature selection improves the detection rate. However, as more features are filtered, detecting large passages becomes difficult. The results presented in Figure 19 shows that when no feature selection is used (threshold = 0), we obtain the highest recall for passages with 50 words. However, as more terms are filtered, the recall for passages with 50 words drops below the recall for passages with 30 and 40 words. Thus, as more terms are filtered, the recall for finding larger passages (50-word) decreases faster than the smaller passages (30-word or 40-word). As mentioned before, the information in larger passages is scattered and vice versa it is dense in smaller passages. Hence, as the threshold increases, more terms are filtered from the feature set. In Figure 19, *overlapping window* approach with window size of 25 is used. The passages with 50 words are divided into two 25-word windows. A 25-word window with passage of 50 words contains less information than a 25-word window for passage of 30-words, as the information in 30-word passage is denser than information in 50-word passage. Hence, as the terms are filtered, finding passages with 50 words is more difficult than finding a passage with 30 words. Thus, the recall of large passages drops faster than in smaller passages.

Figure 19. Behavior of passage detection algorithm on passages of different sizes for different thresholds using overlapping window technique (25 word window)



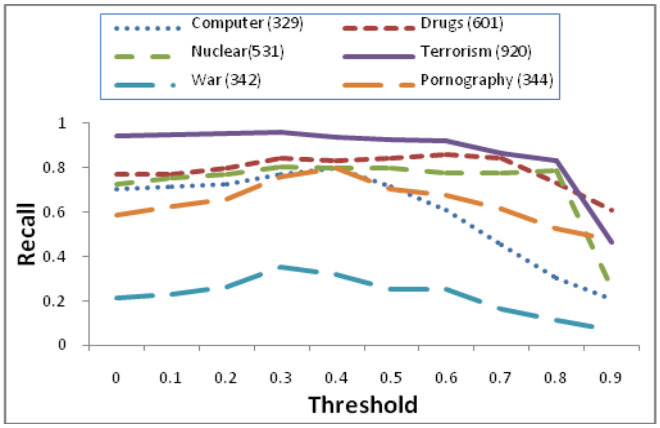
5.3 Effects of topic model on passage category prediction

We demonstrate the effects of the nature of training data on the passage category prediction. We are interested to find the prediction rate of the documents that contain a passage of a given category. Hence, all the values discussed in this section are also recall values. Figure 20 presents the recall of each individual category with respect to different threshold values for overlapping approach with a 25-word window.

It is observed that if more training documents are used for training a category, there is a higher probability of predicting the passages related to those categories. Categories like *Terrorism (920), Nuclear Weapons (531)* and *Drugs (601)* have the most documents in training set and thus are predicted with a higher recall. However, category like war (342), that has the least number of training documents is predicted with a very low recall.

Hence, the recall of passage category prediction for a given category is directly dependent on the number of documents present in the training set of that category. On further analysis, it was found that when the passage actually belonged to category *war*, it was mostly (83% times) misclassified as *terrorism*. As the passages are extracted from CNN news articles from recent years, most of the articles belonging to category *war* are related to ongoing wars in Iraq and Afghanistan that in such news articles

Figure 20. Recall values of each category with respect to different threshold values in overlapping window approach. Values given adjacent to category names in parenthesis are the number of training documents used for each category.



were associated to *terrorism*. Hence, if there are related categories (like *war* and *terrorism*) and one of those categories (*terrorism*) has more training data, it may adversely affect the passage category prediction recall of other category (*war*). We plan to apply our algorithm [12], which discovers the relationships among categories, to find passages with categories that are related to the category of user's interest.

6. CONCLUSION

We developed and evaluated a model for passage detection using text classification and various techniques to split documents into passages. We used a modified version of 20 Newsgroups dataset where passages related to "security" topics are inserted into some documents. We simulated the task of detecting such hidden passages in documents. Our results indicate that as the passage window size increases, precision of detection increases while recall decreases. We compared the effectiveness of different document splitting techniques and found that *overlapping* window approach statistically significantly outperforms other approaches on modified 20 Newsgroups dataset. We also analyzed the effects of different window sizes and feature selection for detecting passages of different lengths. We observed that as the size of window in window passage approaches increases, smaller passages are ignored and larger passages are detected more effectively than smaller passages. Thus, a user needs to decide the size of passages he/she wants to detect, before setting the size of the window. Also, we observed that smaller threshold should be set in feature selection algorithm for finding larger passages and vice versa.

7. ACKNOWLEDGEMENT

We like to thank Shizhu Liu and Malkeet Singh in helping to create the security dataset and inserting the passages into documents to create the testing set.

8. REFERENCES

1. Callan J.P., Passage Retrieval Evidence in Document Retrieval. In Proceedings of the 17th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1994) Pg: 302–310
2. Hazel O., Email and Internet Monitoring in the Workplace: Information Privacy and Contracting-out, Industrial Law Journal, Volume 3, 2002. Pg: 321-352
3. Hearst M., Plaunt C., Subtopic Structuring for Full-length Document Access. In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1993) Pg: 59–68
4. Hearst M., Multi-paragraph segmentation of expository text. In Proceedings of the 32nd. Annual Meeting of the Association for Computational Linguistics, New Mexico, 1994. Pg: 9-16
5. Kaszkiel M., Zobel J., Passage retrieval Revisited. In Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 1997) Pg: 178 – 185
6. Kaszkiel M., Zobel J., Effective ranking with arbitrary passages. Journal of the American Society for Information Science and Technology, Volume 52 Number 4, 2001. Pg: 344 - 364
7. Kim J, Kim M.H., An Evaluation of Passage-Based Text Categorization. Journal of Intelligent Information Systems Volume 23, Number 1, 2004. Pg: 47 – 65
8. Lang K., Original 20 Newsgroups Dataset. http://people.csai.mit.edu/jrennie/20Newsgroups.
9. Ma L., Goharian N., Chowdhury A., Chung M., "Extracting Unstructured Data From Template Generated Web Documents", Proceedings of the ACM 12th Conference on Information and Knowledge *Management (CIKM* 2003).
10. Mengle S., Goharian N., Platt A. FACT: Fast Algorithm for Categorizing Text. In Proceedings of the IEEE 5th International conference on Information and Security Informatics (IEEE ISI 2007) Pg: 308 - 315
11. Mengle S., Goharian N. Using Ambiguity Measure Feature Selection Algorithm for Support Vector Machine Classifier. In Proceedings of the ACM 23rd Annual Symposium on Applied Computing (SAC 2008) Pg: 920 – 925
12. Mengle S., Goharian N., Platt A. Discovering Relationships among Categories using

Misclassification Information. Proceedings of the ACM 23rd Annual Symposium on Applied Computing (SAC 2008). Pg: 932 – 937

13. Zhou W., Yu C., Smalheiser N., Torvik V, Jie H., Knowledge-intensive Conceptual Retrieval and Passage Extraction of Biomedical Literature. Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007). Pg: 655 – 662