Passage Detection using Text Classification

Saket Mengle

Information Retrieval Lab, Illinois Institute of Technology, Chicago, Illinois, U.S.A., saket@ir.iit.edu

Nazli Goharian

Information Retrieval Lab, Illinois Institute of Technology, Chicago, Illinois, U.S.A., nazli@ir.iit.edu

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 organizations. Passage retrieval is well studied; we posit, however, that passage detection is not. Passage retrieval is the determination of the degree of relevance of blocks of text, namely passages, comprising a document. Rather than determining the relevance of a document in its entirety, passage retrieval determines the relevance of the individual passages. As such, modified traditional information retrieval techniques compare terms found in user queries with the individual passages to determine a similarity score for passages of interest.

In passage detection, passages are classified into predetermined categories. More often than not, passage detection techniques are deployed to detect hidden paragraphs in documents. That is, to hide information, documents are injected with hidden text into passages. Rather than matching query terms against passages to determine their relevance, using text mining techniques, the passages are classified. Those documents with hidden passages are defined as infected. Thus, simply stated, passage retrieval is the search for passages relevant to a user query, while passage detection is the classification of passages. That is, in passage detection, passages are labeled with one or more categories from a set of predetermined categories.

We present a *keyword based dynamic passage* approach (KDP) and demonstrate that KDP outperforms statistically significantly (99% confidence) the other document splitting approaches by 12% to 18% in the passage detection and passage category prediction tasks. Furthermore, we evaluate the effects of the feature selection, various passage lengths, ambiguous passages, and finally training data category distribution on passage detection accuracy.

Introduction

Traditionally, text classifiers are used to identify the category 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 passages whose contents differ from the category assigned to that document by a text classifier. For example, consider the following document, which is a paragraph from CNN website (www.cnn.com) with a passage inserted within the text:

The volume increased after Federer lost the French Open and Wimbledon finals. From all around the world they arrived, some to his parents' house in Switzerland, some to his agent, some to his hotels. They came from retired players and from current coaches, from doctors, from fans. They offered good wishes, medical advice, even tennis advice. ***Lehman Brothers investment bank announces it's filing for bankruptcy.*** *Everyone figured Federer needed help, and everyone figured they knew how to help. Turns out Federer was just fine. Turns out he still knew how to win a major tournament. He proved that Monday night, easily beating Andy Murray 6-2, 7-5, 6-2 to win a fifth consecutive U.S. Open championship and 13th Grand Slam title overall.*

The text seems to be about *Sports*. However, one of the lines in the text is about breaking news in *Finance*. Though text classifiers work effectively to identify the category of a document as whole, they fail to identify the category of such hidden passages. If one is interested to detect any passage about *Finance*, the highlighted passage is not detected during the process of text classification. Finding such passages is critical during the process of detecting insider misuse if an insider is trying to leak *Financial* information using a document about *Sports* as a wrapper.

Insider misuse is discussed in the context of the detection of hidden text within e-mail messages (Hazel, 2002). However, documents are typically longer than e-mail messages; thus, text classifiers are either inefficient or even possibly unable to detect hidden passages within lengthy documents.

Another application for this research is routing passages from documents that match the users' interest category. Within this context, text classification is commonly used,

and the document, in its entirety, is categorized and routed to a user. A document that matches the user specified category or categories is routed correspondingly. The problem with such a solution is that a passage from a document, which is not identified as a matching category, may be still of interest to the user.

Passage retrieval research efforts (Callan, 1994), (Kaszkiel & Zobel, 2001) have addressed approaches to find passages in a document that match a user's query. However, passage retrieval approaches do not identify the passages based on the category of a passage, but exploit the user queries. Hence, if one wants to find information related to *Finance* in the above passage, he/she needs to submit all possible queries related to *Finance* to find the highlighted passage.

We propose *passage detection* that detects the passages related to a given category rather than related to queries. Unlike passage retrieval (Grossman & Frieder, 2004), passage detection uses supervised learning to train a text classification model. This classification model is used to classify passages in documents and identify a document if it contains passages related to the category of user's interest.

We present a three-phase methodology for passage detection (Mengle & Goharian, 2008a). 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. We also apply two feature selection techniques to filter unimportant terms from the trained model. In the second phase, we preprocess the documents by dividing a document into passages using various document splitting techniques. In the third phase, the text classification model is used to detect the *infected* documents. Infected documents are the documents that contain a passage(s) that belongs to the category of user's interest.

We use four variations of Reuters 21578 and 20 Newsgroups datasets to evaluate our approaches. We evaluate our passage detection approaches for three tasks, namely, passage detection (*PD*), stringent passage category prediction (*S-PCP*) and tolerant passage category prediction (*T-PCP*). *PD* only detects the presence of a user specified category in a document. *S-PCP* and *T-PCP* predict the categories of such passages. However, *T-PCP* allows a classifier to also predict categories related to the actual category even if not the exact category. Details about our evaluation tasks are given in the evaluation metrics section.

We empirically demonstrate that *keyword-based dynamic passage* approach outperforms statistically significantly (99% confidence) the other document splitting approaches in all three tasks. Furthermore, our empirical results indicate that using feature selection statistically significantly (99% confidence) improves the effectiveness of passage detection algorithms. Our results also indicate that as the window size increases, the probability of detecting small passages decreases. Furthermore, we show that as the number of infected passages in testing documents increases, the effectiveness of detecting such passages in documents also increases. Finally, we evaluate the effects of category distribution in training documents on passage detection.

Prior Work

Passage retrieval is the task of retrieving only the portions of documents that are relevant to a particular information need (Wade and Allen, 2005). Various methods are applied for finding relevant passages among documents.

Traditionally, term frequency/inverse document frequency (*tfidf*) was used for passage retrieval. Each passage and query is modeled as a vector in the space of all the terms in a dataset. The score of a passage for a given query is calculated as the inner product of the vector representing the passage and the vector representing the query. A normalized *tfidf* formula that considers length of passages is presented in (Allan et al., 2003).

The *Query Likelihood Language Model* (Ponte and Croft, 1998) uses statistical language model for passage retrieval. The probability of each passage in a document with respect to a given query is considered to predict if a passage belongs to that query. A Multiple-Bernoulli model was used to estimate the probability that a given query is relevant to a given passage. A similar model is also used in (Hiemstra & Kraaij, 1998), (Miller, Leek & Schwartz, 1999), (Song & Croft, 1999).

The relevance models in (Lavrenko & Croft, 2001) provide a language modeling based approach for estimating a probability for each word in the relevant class of documents. Using relevance model, they retrieve documents that contain the query words and also the documents that are relevant to the topic of query words. (Wade and Allen, 2005) combines the relevance model with the maximum likelihood model of the original query to place higher weights on the original query terms. This smoothed relevancy model is applied to find relevant passages among documents with respect to a given query.

Text classification algorithms such as Support Vector Machine (SVM) are also used in passage retrieval. The retrieved results are marked as relevant (positive samples) for a given query and all other documents in the dataset are marked as negative samples. The data are used as training information to train a SVM classification model. This model is used to classify passages as relevant or not relevant to that given query. However, one of the problems

Input:

a) User specified categories.

b) Documents for training the text classifier containing documents that are labeled with both the categories of user's interest and categories that user is not interested in.

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 categories as well as other categories. That is for each document term the AM value is calculated.

Phase II

a) Parse the input documents to be tested.

b) Split each document into passages using a document splitting technique.

Phase III

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

b) 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.*

Figure 1. Pseudocode of the three phase methodology for passage detection

with applying SVM for passage retrieval task is that the class of negative samples is significantly larger than the class of positive samples. To counter this problem, only few negative samples are randomly selected (Nallapati, 2004), or a *bootstrap* method is used (AbdulJaleel et al., 2004).

A model for classifying passages was proposed in (Kim & Kim, 2004). However, the objective was to classify a document in its entirety using passage category information. Our objective is to correctly identify the passages based on user specified category.

As passages are located at random locations in documents, identifying the boundaries of passages is critical. Various techniques are used to split a document for passage retrieval. (Callan, 1994), (Salton, Allan & Buckley, 1993), (Salton, Allan & Singhal, 1996) assume that the boundary of a passage is predefined based on discourse information in a passage. The effort in (Zhou et al., 2007) assumes that <*p*> and </*p*> HTML tags mark the start and the end of a passage, respectively. The discourse information such as a sentence or group of sentences is also used in (Hearst & Plaunt, 1993). However, there are a few drawbacks in using the discourse information to define passages. First, there may be discourse inconsistency among authors. Second, it may be impossible to create discourse passages, if the discourse information such as punctuation marks or HTML tags is not provided with a document (Kaszkiel & Zobel, 1997). Finally, the discourse passages can be very small or very large based on the author's style. Windowing approaches are also used to dynamically identify passage boundary based on the particular query term (Callan, 1994). Each passage consists of the same number of words. However, the drawback is the unknown passage length. If the window size is too small, larger passages are not detected, and if the window size is too large, smaller passages are not detected. Hence, we present a new technique called *keyword-based dynamic passage* (*KDP*) approach that takes advantage of the term weights to detect passages. This method is independent of passage lengths and takes advantage of the information generated by the text classifiers.

Methodology

Our objective is to predict if a document contains passages about a category that a user is interested in. Figure 1 presents the pseudocode of the three phases of our detection methodology. Details of each phase follow.

Phase I: Building a text classification model

In the first phase, a text classification model is built based on the training documents. We used a Naïve Bayes classification algorithm to build our passage detection model. The Naïve Bayes classifier is a multinomial classifier and is suited for domains where users are interested in multiple topics. Although SVM is shown to be more effective than Naïve Bayes classifier (Joachims, 1998), SVM is a binary classifier, and hence, its effectiveness depends on the distribution of positive and negative samples. However, in our application, the number of training documents that belong to the category of user's interest (positive samples) are comparatively fewer than negative samples. Furthermore, the training of Naïve Bayes classifier is in linear time, unlike in SVM. We improved the effectiveness of the model by using two feature selection algorithms, namely Odds ratio (Mladenić &

Grobelnik, 1998) and Ambiguity Measure (AM), which was shown to outperform the existing feature selection algorithms (Mengle, Goharian, 2008b). We evaluated the effectiveness of these feature selection algorithms on *unbalanced* datasets and observed that AM is better suited for such tasks. The nature of unbalanced datasets is such that few categories have significantly more training data than others. This leads to a higher term frequency of features in these categories. Although these features may point to more than one category, *odds ratio* assigns a higher weights to them. *AM* assigns a high weight to a term, if it appears consistently in only one specific category. The intuition is that the term that appears in only one category points stronger to that specific category, and thus, is a better indicator in a classification decision. A brief explanation about the two feature selection algorithms follows.

Odds ratio: The basic idea of using *odds ratio* (Mladenić & Grobelnik, 1998) is to calculate the odds of a term occurring in the positive class (the category a term is related to) normalized by the odds of that term occurring in the negative class (the category a term is not related to). The *odds ratio* of a term t_k for a category c_i is defined using Formula 1:

$$Odds\ Ratio\ (t_k, c_i) = \frac{P(t_k \mid c_i)[1 - P(t_k \mid \overline{c_i})]}{[1 - P(t_k \mid c_i)]P(t_k \mid \overline{c_i})} \quad ..1$$

Odds Ratio is known to work well with the Naive Bayes text classifier algorithm (Mladenić et al., 2004).

Ambiguity Measure: Ambiguity measure (AM) (Mengle and Goharian, 2008) assigns a high score to a term, if it appears consistently in only one specific category. *AM* for a term t_k with respect to category c_i is calculated using Formula 2. The maximum *AM* score for term t_k with respect to all categories is assigned as the *AM* score of term t_k (Formula 3).

$$AM(t_k, c_i) = \left(\frac{tf(t_k, c_i)}{tf(t_k)} \right) \quad ... 2$$

$$AM(t_k) = \max(AM(t_k, c_i)) \quad ... 3$$

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.

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 AM of each term that occurs in training documents is calculated.

Phase II: Splitting algorithms

In the second phase, the testing documents are divided into passages based on various document splitting approaches. A passage is defined as any sequence of text within a document (Kaszkiel & Zobel, 1997). Various types of automatic document splitting techniques exists, each of which defines a passage differently as described in the prior work section.

We introduce a document splitting method called *keyword-based dynamic passage* approach and compare its effectiveness with three document splitting approaches, namely *discourse passage* approach, *non-overlapping window passage* approach and *overlapping window passage* approach. A detailed explanation about these document splitting approaches are as given below.

Keyword-based Dynamic Passage (*KDP*)

Prior efforts in document-splitting approaches did not use the information pertaining to the document categories. However, in text classification, feature selection algorithms assign a weight to each document term to indicate the strength of relevance of a term to a given category.

In the *keyword-based dynamic passage (KDP)* approach (Goharian & Mengle, 2008) passages are defined around terms with higher weights. We assume that the probability of detecting the correct category of a passage is higher when the passage contains a term with a higher weight. We call these terms *keywords*. Thus, for a fixed length passage with *n* words, we define a passage from *n/2-1* terms before a high-weight term and up to *n/2* terms after that term. Hence, we guarantee that each passage has at least one term with a higher weight. Formulae 4 and 5 are applied for defining the start and end of the passage, where the weight of a term t_k is higher than an empirically determined term weight threshold. We determined this threshold exhaustively (0.2) to maximize the F1 measure for passage detection.

$$Start\ (Passage) = Position\ (t_k) - (\frac{n}{2} - 1) \quad ..4$$

$$End\ (Passage) = Position\ (t_k) + (\frac{n}{2}) \quad ..5$$

Discourse Passage (*DP*)

Discourse passages (DP) are based on logical components including discourse boundaries such as a sentence or a paragraph (Callan, 1994). An example of *DP* approach is shown in Figure 2. In this example, a document is split into three passages such that each passage contains

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

Figure 2. Example of *DP* where a document is divided into passages based on sentence boundaries (n=1)

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

Figure 3. Example of *NWP* approach where each passage has same number of words (n=4)

one sentence. In our experiments, we consider *passages* as a group of *n* sentences. We determined the value of *n* based on maximizing the detection accuracy in terms of F1 measure. For that we evaluated *n* for 1 to 5 sentences.

Non-overlapping Window Passage (*NWP*)

Unlike the *DP* approach where passages are determined based on the structural properties of the document, the *window passage* approach defines a passage as *n* number of words. (Hearst, 1994) proposes the *non-overlapping window passage* (*NWP*) approach where documents are segmented into evenly sized blocks. An example of *NWP* approach is shown in Figure 3. There is no shared area between two adjacent windows, and hence, these windows are called *non-overlapping* windows. We evaluated the effect of window sizes of 5, 10, 15, 20, and 25 words.).

Overlapping Window Passage (*OWP*)

The *NWP* approach may break a passage that relates to a user specified category into two passages. For example, if the size of window is ten words and the size of an *infected* passage is also ten words, then the worst case would be that five words from the infected passage are used in one window and the other five words are used in another window. In this case, both these windows may contain words that do not logically belong to that infected passage. Thus, the classification accuracy decreases. To avoid such situations, (Callan, 1994) proposed the concept of overlapping windows. In the *overlapping window passage (OWP)* approach, a document is divided into passages of evenly sized blocks by overlapping *n/2* from the prior passage and *n/2* from the next passage.

In Figure 4, we show an example of *OWP* approach. Similar to the *NWP* approach, we evaluated the effect of various window sizes on passage detection.

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	

Figure 4. Example of the *OWP* approach where each passage has same number of words and windows are overlapped (n=4)

Phase III: Classifying passages

The classification model built in *Phase I* is used for individually classifying each passage that was identified based on the document splitting techniques in *Phase II*.

As we use feature selection algorithms, all terms with low AM/Odds Ratio weight are pruned from the feature set. Hence, although a passage is small, the classification of that passage is based on the high weight terms that exist in that passage. This reduces the number of false positives generated during passage detection.

Based on the classification results, if a document contains an *infected* passage, we mark that document as *infected*.

Experimental Setup

We explain our experimental framework and datasets used to train and test our classification model.

Dataset

To validate our passage detection effectiveness, we needed a dataset that contained documents belonging to a given category and were *infected* with passages of a different category. To our knowledge, no such dataset is available. Hence, we used the 20 Newsgroups (20NG) and the Reuters 21578 datasets. These datasets contain news articles about various topics such as sports, electronics, science, politics, religion, etc. We then inserted passages related to *security* subjects that are extracted from news articles on the CNN web site into some documents from 20NG and Reuters 21578. These established our modified set of 20NG and Reuters 21578 documents. Documents from the 20NG dataset and Reuters 21578 dataset were used to train a text classifier on various existing categories. A set of news articles were crawled from CNN website to train the classifier on categories of the *infected* passages. In the testing phase, both *infected* and non-infected (*clean*) documents were used. As 5% of the documents from 20NG and Reuters are infected, we used a 9-1 split such that half of the testing documents were infected and the other half is not infected. Details on each of our datasets (training and testing) follow.

Training Documents

We used 20 Newsgroup (20NG) or Reuters 21578 datasets to train the text classifier to detect passages that are related to categories present in 20NG or Reuters 21578 dataset, respectively. Moreover, to train the text classifier on topics related to *security*, we created a dataset that contains documents related to *security* subjects. Details about these datasets follow.

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	Modified 20 Newsgroups dataset				Modified Reuters 21578 dataset			
	Dataset	Number of documents	Document infected?	Passage Length	Dataset	Number of documents	Document infected?	Passage Length
Training	20 NG	18,000	-	-	Reuters 21578	9900	-	-
	Security Dataset	3067	-	-	Security Dataset	3067	-	-
Testing	20 NG	1000	No	-	Reuters 21578	550	No	-
	20 NG	200	Yes	10 words	Reuters 21578	110	Yes	10 words
	20 NG	200	Yes	20 words	Reuters 21578	110	Yes	20 words
	20 NG	200	Yes	30 words	Reuters 21578	110	Yes	30 words
	20 NG	200	Yes	40 words	Reuters 21578	110	Yes	40 words
	20 NG	200	Yes	50 words	Reuters 21578	110	Yes	50 words

20 Newsgroups

The 20 Newsgroup[1] corpus consists of a total of 20,000 documents that are categorized into twenty different categories. Each category contains 1,000 documents. The average document length in the 20NG dataset is 311 (non-unique) terms per document. Some of the newsgroups categories are very closely related to each other (e.g., comp.sys.ibm.pc.hardware and comp.sys.mac.-hardware), while others are highly unrelated (e.g., misc.forsale and soc.religion.christian). This characteristic contributes to the difficulty of categorization of documents that belong to very similar categories.

Reuters- 21578 Dataset

The Reuters 21578[2] corpus contains the Reuters news articles from 1987. These documents range from multi-labeled, single labeled, or not labeled. The average document length in Reuters 21578 dataset is 200 (non-unique) terms per document. Thus, the average size of the documents is smaller than those in 20 Newsgroups dataset. Reuters 21578 dataset consists of a total number of 135 categories (labels), ten of which have significantly more documents than the rest of the categories. Thus, commonly the top 10 categories are used to evaluate the accuracy of the classification results. The top 10 categories of Reuters 21578 are "earn", "acq", "money-fx", "grain", "trade", "crude", "interest", "wheat", "corn" and "ship".

Security Dataset

We created a dataset related to *security* subjects 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 (Ma, Goharian & Chowdhury, 2003), 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* for each of the six categories. Before performing evaluation, the evaluators agreed upon the definition of each category. The average Pearson's co-relation between the assessor's evaluations was 90.60%.

[1] Lang K., Original 20 Newsgroups Dataset. http:// people.csai.mit.edu/jrennie/20Newsgroups

[2] Lewis D., Reuters-21578, http://www.daviddlewis.com/ resources/testcollections/reuters21578.

Testing documents

We created four variations of each dataset by inserting one to four passages into half of the testing documents. To observe the effects of our algorithm on passages of varying length, we inserted passages of 10 words, 20 words, 30 words, 40 words and 50 words. Every passage is inserted at a random location in a document. Discourse boundaries such as HTML tags are filtered out of the passages that are inserted. Each inserted passage is evaluated by two graduate students to confirm that it belongs to one of the categories in the security dataset. Table 2 shows statistics of the testing documents based on the modified 20NG and Reuters 21578 datasets with respect to the presence of passages related to security subjects and the length of such passages.

Evaluation Metrics and Tasks

We define various evaluation tasks for passage detection. In the first task, called *Passage Detection*, we evaluate the effectiveness of detecting the presence of *infected* passages in a document. In the second task, called *Stringent Passage Category Prediction*, we evaluate if the category of the infected passage is detected correctly. In the third task, called *Tolerant Passage Category Prediction*, we allow the classifier to assign categories to infected passages such that these categories are either the actual categories of those passages or are the categories that are related to the actual categories. Details about these evaluation tasks follow.

Passage Detection (PD) Task

In *passage detection*, true positive is generated for all instances where a document contains an *infected* passage and the classifier marks the document as *infected* (Table 3).

Stringent Passage Category Prediction (S-PCP) Task

In *Stringent Passage Category Prediction* (S-PCP) (Table 4), true positives are generated for instances where the classifier correctly predicts the category of the *infected* passages, that is, when the category of the detected passage is exactly as the actual category.

Tolerant Passage Category Prediction (T-PCP) Task

Furthermore, we evaluate the accuracy of the prediction of passage category by additionally considering *related* categories to the actual category. We call the task *Tolerant Passage Category Prediction* (*T-PCP*) (Table 5). This generates true positives for instances where the predicted category is either the actual category or is *related to* the actual category of a passage, even though it is not exactly the same category. We automatically find relationships among categories using the technique presented in (Mengle, Goharian & Platt, 2008), which used misclassification information from a text classifier to find relationships among the categories that are present in the dataset. *T-PCP* performs significantly better than *S-PCP* as it predicts either the correct category or related category.

To evaluate the effectiveness of our approach, we use the commonly used evaluation metrics: precision, recall and F1 measure. Precision is defined as the ratio of infected documents detected correctly to the number of documents predicted as infected (Formula 6). Recall is defined as the ratio of infected documents detected correctly to the total number of infected documents available in the testing set (Formula 7). F1 measure is the harmonic mean of precision and recall (Formula 8).

$$\text{Precision (P)} = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad .. 6$$

$$\text{Recall (R)} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad .. 7$$

$$\text{F1measure} = \frac{2PR}{P+R} \quad .. 8$$

Table 3. Contingency matrix for *PD*

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

Table 4. Contingency matrix for *S-PCP*

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

Table 5. Contingency matrix for *T-PCP*

Actual \ Predicted		Infected: Passage with category x	Infected: Passage with category related to x	Infected: Passage with category not related to x	Not Infected
Infected	Passage with category x	TP	TP	FN	FN
	Passage with category not related to x	FP	FP	TN	
Not Infected		FP			TN

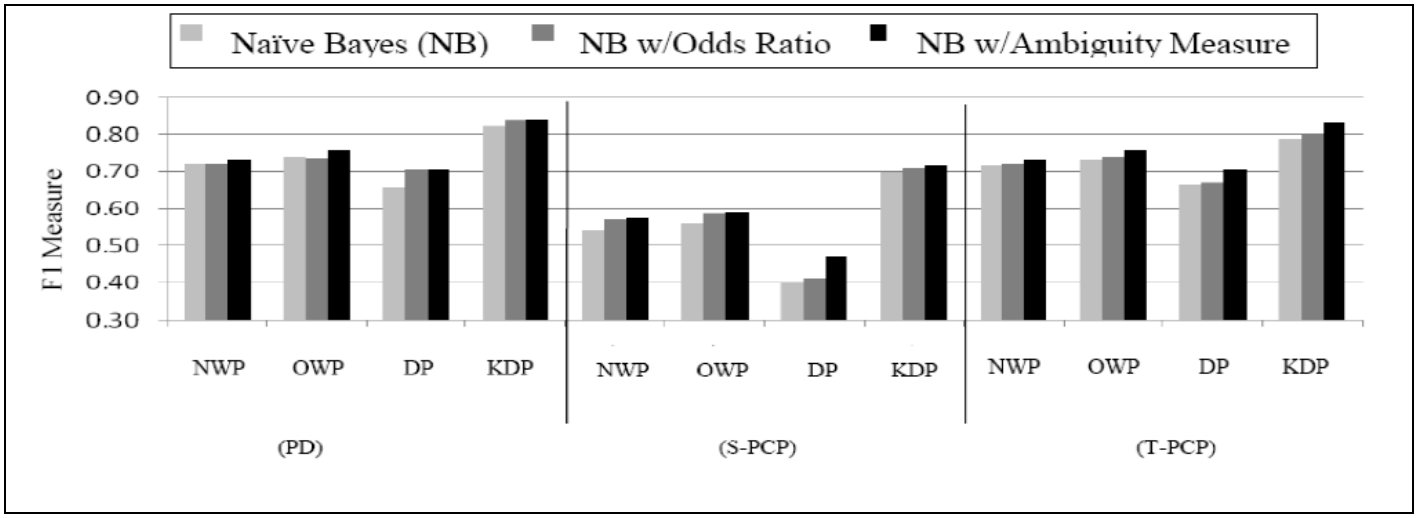


Figure 5. Effects of various document splitting approaches and feature selection algorithms for each of the three evaluation tasks using 20NG dataset

Results

Table 6 provides a list of all the acronyms and their descriptions.

Table 6. Acronym Table

Acronym	Description
NWP	Non-Overlapping Window Passage
OWP	Overlapping Window Passage
DP	Discourse Passage
KDP	Keyword-based Dynamic Passage
PD	Passage Detection
S-PCP	Stringent Passage Category Prediction
T-PCP	Tolerant Passage Category Prediction

We present the following results:

- Comparison of the four document splitting techniques for each of the three evaluation tasks.
- Effects of feature selection on the passage detection process.
- Effects of window size in windowing approaches (NWP and OWP) and keyword weight thresholds in KDP.
- Effects of the passage length on the detection rate (recall).
- Effects of the varying number of passages in a document on F1 measure.
- Effect of document category distribution on the passage category prediction recall.
- Effects of presence of ambiguous passages in documents on F1 measure.

Comparison of Document Splitting Approaches

We evaluate and compare the four document splitting approaches (KDP, DP, NWP, OWP) for each of the three evaluation tasks (PD, S-PCP, T-PCP) using the 20NG and Reuters 21578 datasets. Figures 5 and 6 depict this comparison. The X-axis represents the various document splitting approaches, and the Y-axis represents the F1 measure. We observed that the *keyword-based dynamic passage (KDP)* approach statistically significantly (99% confidence) outperforms the *overlapping window passage (OWP)* approach (which is the second best performing document splitting approach) in *PD* (by up to 12%), in *S-PCP* (by up to 18%) and in *T-PCP* (by up to 12%) with respect to the F1 measure. Also, it can be noted that the *OWP* approach performs significantly better than the *non-overlapping window passage* (*NWP*) approach with respect to the F1 measure in *PD* (by up to 2.7%), in *S-PCP* (by up to 5.1%) and in *T-PCP* (by up to 2.8%).

For the *KDP* approach, as we only detect passages that contain terms with higher *AM* weight (i.e., less ambiguous terms), the number of false alarms significantly decreases and hence, the precision increases. As we define a new passage around each keyword, the probability of detecting *infected* passages increases. Thus, the recall value also increases. Consequently, the F1 measure for KDP is significantly better than other approaches.

The *NWP* approach may have some degree of loss of information since an infected passage may be split and become part of adjacent windows. The *OWP* approach avoids such loss of information since it also generates passages that overlap with adjacent passages. Hence, the *OWP* approach, as shown, performs statistically significantly better (by up to 5.1%) than the *NWP* approach

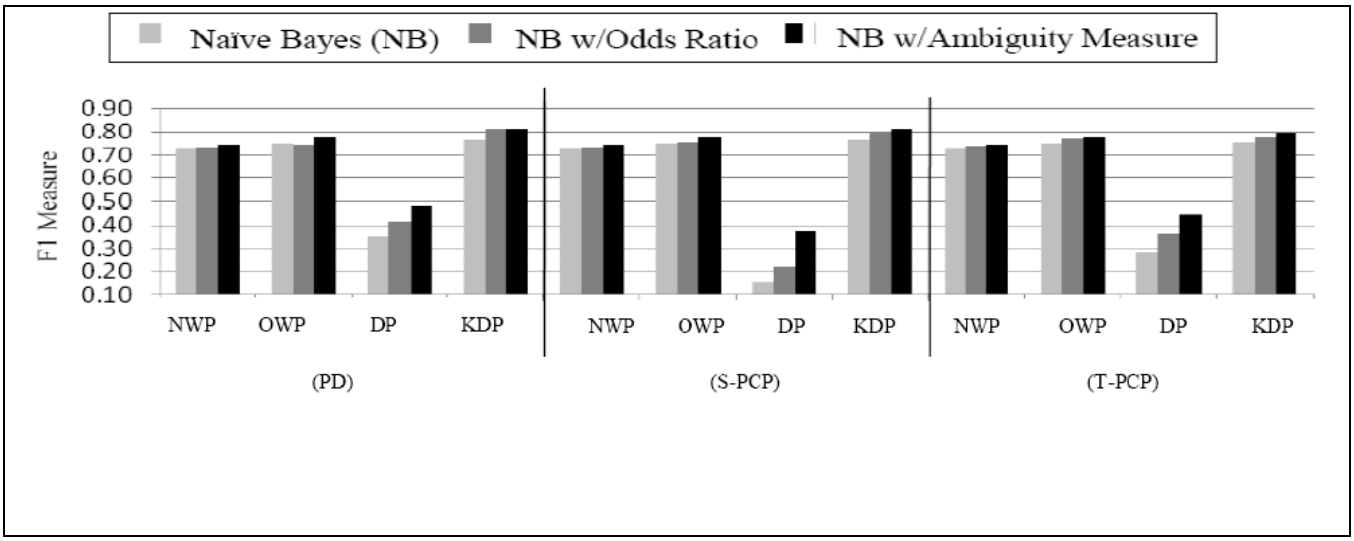
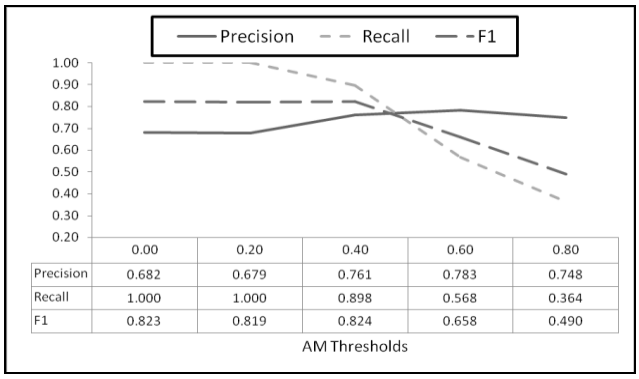


Figure 6. Effects of various document splitting approaches and feature selection algorithms for each of the three tasks using Reuters 21578 dataset



	0.00	0.20	0.40	0.60	0.80
Precision	0.682	0.679	0.761	0.783	0.748
Recall	1.000	1.000	0.898	0.568	0.364
F1	0.823	0.819	0.824	0.658	0.490

Figure 7. Effects of AM thresholds on 20NG dataset

with respect to the F1 measure. The *Discourse Passage (DP)* approach performs statistically significantly worse than both *NWP* (9.2%) and *OWP* (12.9%) approaches. As mentioned in the experimental setup section, the discourse information such as delimiters and passage tags were removed from the inserted passages to simulate realistic scenario as to malicious cases. Hence, detecting passages that do not contain discourse information is difficult.

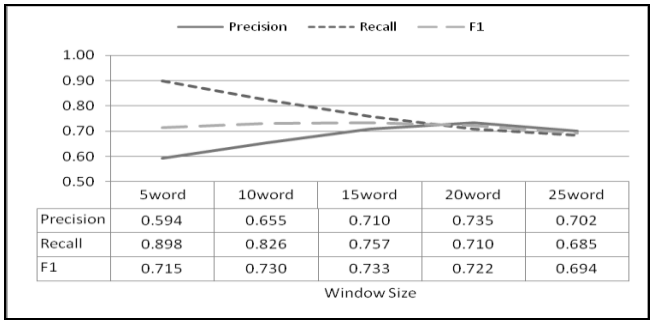
Effects of Feature Selection

Figures 5 and 6 also depict the effects of *odds ratio* and *AM* feature selection algorithms on the three passage detection evaluation tasks (PD, S-PCP, T-PCP). Using feature selection significantly (99% confidence) improves the effectiveness of *PD* (by up to 8%), *S-PCP* (by up to 12%) and *T-PCP* (by up to 14%) with respect to the F1 measure. Our results also indicate that *AM* performs significantly better than the *Odds Ratio* feature selection algorithm. Feature selection prunes words with a lower term weight from the feature set of a text classifier and only keeps the most important terms. 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. This leads to an improvement in the F1 measure.

Figure 7 shows the trends in precision, recall and F1 measure with respect to various *AM* feature selection thresholds. As shown, precision consistently increases for increasing value of *AM* weight threshold, from 68.2% (Threshold: 0.0) to 74.8% (Threshold: 0.8). However, as many of non-discriminating terms (terms with a low *AM* value) are filtered, some infected passages that do not contain keywords are undetected. Hence, the recall of passage detection decreases from 100% (Threshold: 0.0) to 36.4% (Threshold: 0.8) when the AM threshold increases. Nevertheless, as indicated by the results, feature selection significantly improves precision and F1 measure.

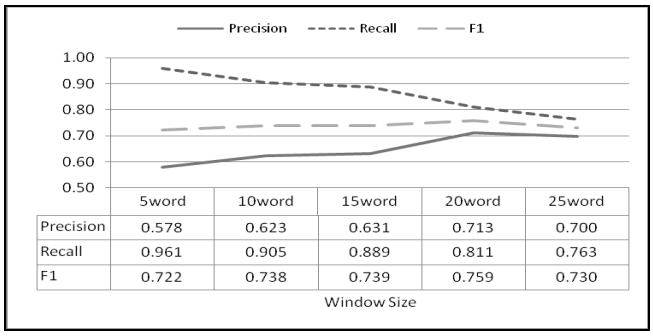
Effects of window sizes on windowing approaches

Figure 8 illustrates the effects of window size on the *non-overlapping window* (*NWP)* approach for the *passage detection* (*PD)* evaluation task using the 20 Newsgroups dataset. The X-axis indicates different window sizes (5, 10, 15, 20, and 25 word) that were used for experimentation. For larger window sizes, the classifier uses more words for classification. As the window size increases, the precision of *PD* also increases. The precision of the *NWP* approach increases from 59.4% (for 5-word window) to 73.5% (for 20-word window). However, if the window size is very large, smaller passages are not predicted correctly, and hence, precision may drop. As shown, precision dropped by 4.1% when the window size changed from 20 words to 25 words.



	5word	10word	15word	20word	25word
Precision	0.594	0.655	0.710	0.735	0.702
Recall	0.898	0.826	0.757	0.710	0.685
F1	0.715	0.730	0.733	0.722	0.694

Figure 8. Effects of window size on NWP approach using 20 Newsgroups dataset for PD evaluation task



	5word	10word	15word	20word	25word
Precision	0.578	0.623	0.631	0.713	0.700
Recall	0.961	0.905	0.889	0.811	0.763
F1	0.722	0.738	0.739	0.759	0.730

Figure 9. Effects of window size using OWP approach in 20 Newsgroups dataset for PD evaluation task

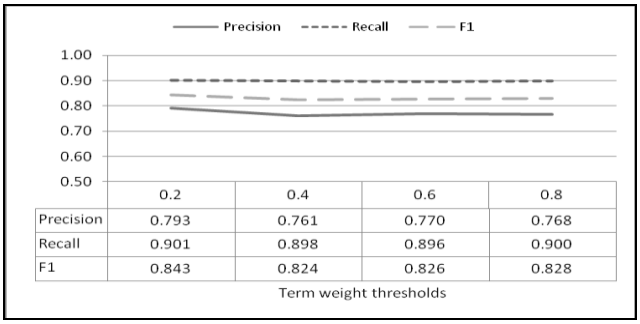
Furthermore, when the window size is large, smaller passages that are present in a document are not detected, resulting in a decrease in recall, from 89.8% (for 5-word window) to 68.5% (for 25-word window).

Figure 9 demonstrates that the *OWP approach* follows similar trends as the *NWP approach* for different window sizes. Similar trends are also observed for both the *S-PCP* and *T-PCP* evaluation tasks, and also on the Reuters 21578 dataset.

Effects of Keyword Threshold for KDP approach

The trends for the *KDP* approach are shown in Figure 10. Defining *keywords* is an important issue in the *KDP* approach. Hence, we evaluate various threshold values such that passages are only generated around terms with the term weight values above that threshold. We call this threshold *term weight threshold*. The X-axis in Figure 10 represents the term weight threshold.

Recall for the PD task is high (90%) and consistent for various *term weight thresholds*. The precision initially decreases when the *term weight threshold* is increased and then increases after a certain threshold. We further discuss the reason for this. Note that we used a threshold of 0.4 for *Ambiguity Measure* feature selection algorithm for the results presented in Figure 10. We call this threshold *AM threshold*.



	0.2	0.4	0.6	0.8
Precision	0.793	0.761	0.770	0.768
Recall	0.901	0.898	0.896	0.900
F1	0.843	0.824	0.826	0.828

Figure 10. Effects of term weight thresholds on KDP approach using 20 Newsgroups dataset for PD task

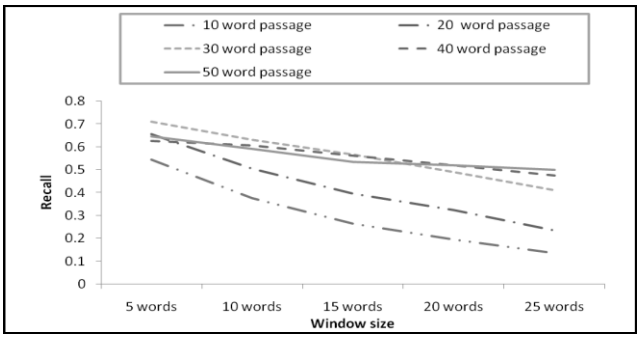


Figure 11. Recall values with respect to various passage lengths for different window sizes on OWP approach using 20 Newsgroups dataset

When the *term weight threshold* is less than the *AM threshold*, passages are formed around keywords whose term weight is lower than the *AM threshold*. Keywords from such passage may be filtered. Such passages are marked as infected only if these passages contain other keywords. As the passage is only classified if it contains important keywords, fewer false positives are generated and hence, the precision of *PD* increases. The precision for the *term weight threshold* of 0.2 is 79.3%.

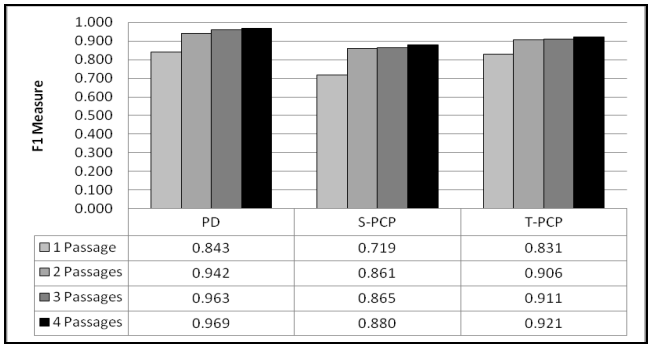
If the *AM threshold* and the *term weight threshold* are set low, passages are generated around unimportant terms. In such cases false positives are generated and hence, the precision of *PD* decreases. The precision for the *term weight threshold* of 0.2 is 76.1%.

When the *term weight threshold* is higher than the *AM threshold*, passages are generated around good keywords (keywords with high term weight). As the passages have at least one good keyword, the precision of *PD* increases. The precision for the *term weight threshold* of 0.2 is 76.8%.

Similar trends are observed for both the *S-PCP* and *T-PCP* tasks and on the Reuters 21578 dataset.

Effects of passage length

We now analyze how the passage length of a document affects the recall (detection rate) of passage detection techniques (Figure 11). In our modified 20 Newsgroups



	PD	S-PCP	T-PCP
1 Passage	0.843	0.719	0.831
2 Passages	0.942	0.861	0.906
3 Passages	0.963	0.865	0.911
4 Passages	0.969	0.880	0.921

Figure 12. Effects of varying number of passages in testing documents on KDP approach using 20 Newsgroups dataset

dataset, passages of varying length (10, 20, 30, 40, and 50 words) were inserted into original documents.

We show our results for the KDP and *OWP* approaches. Similar trends to OWP are also observed for both the *NWP* and *DP* approaches.

The effects of varied passage lengths with different window sizes on the *OWP* approach are shown in Figure 11. The X-axis represents various window sizes, and the Y-axis represents recall values for each run. As shown, for 5-word window, 30-word passage (70.1%) performs significantly better than 50-word passages (63.4%). However, as the size of the window increases, smaller passages are less detected but the detection rate for larger passages is increased. For 25-word window, 50-word passage (49.3%) performs significantly better than 30-word passages (42.1%). This trend indicates that the knowledge about the *infected* passage length is important in selecting the window in the *OWP* approach. Keywords are sparse in larger passages. Hence, it is difficult to detect large passages using small window size of 10 or 20 words. The same observation is noted on the Reuters 21578 dataset.

The recall of the *KDP* approach for detecting passages with the length of 10-word, 20-word, 30-word, 40-word and 50-word, regardless of the window size, are 64.5%, 74%, 87%, 81% and 81%, respectively. In the KDP approach as the windows are defined abound keywords, the recall depends on the number of keywords present in a passage As the number of keywords in 10-word and 20-word passages are lower than in 30-word passages, the recall for detecting 10-word passages is relatively lower. We observed the best recall for detecting 30-word passages. The recall of *KDP* also depends on the density of keywords in a passage. If the density of keywords in a passage is high, then a window around a keyword may consist of multiple keywords. Hence, the recall for detecting 40-word and 50-word passages decreases, as keywords are sparse in larger passages.

Effects of varying number of passages in documents

To observe the effects of varying the number of passages of a document on detection, we created four variations of the testing documents for each dataset. We inserted passages into these four variations of the 20 Newsgroups dataset, namely 20NG-1, 20NG-2, 20NG-3 and 20NG-4 with one, two, three and four *infected* passages, respectively. We similarly created four variations of the Reuters 21578 dataset. We present only the results of the *KDP* approach using 20 Newsgroups dataset to maintain brevity. However, similar trends were observed for other document splitting techniques, both using the Reuters 21578 datasets and 20 Newsgroups datasets.

Figure 12 depicts the effects of varying the number of passages for the variations of 20 Newsgroups dataset for our three evaluation tasks. As the number of infected passages in testing documents increases, the F1 measure for *passage detection* consistently increases, due to a higher probability of detection of at least one passage. It is observed from Figure 12 that the F1-measure for PD task on 20NG-4 (96.9%) is significantly higher than on 20NG-1 (84.3%). Similar trends are also observed for both the *S-PCP* and *T-PCP* evaluation tasks, and also on the Reuters 21578 dataset.

Effects of category distribution in training documents

Figure 13 presents the recall of each category with respect to different AM threshold values for *KDP* approach. We are interested to find the number of passages for each category that are correctly predicted. Hence, all the values discussed in this section are recall values.

Categories such as *Terrorism* (920 training documents), *Nuclear Weapons* (531 training documents) and *Drugs* (601 training documents) have the most number of documents in training set and thus are predicted with a higher recall. However, a category such as *war* (342 training documents) 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

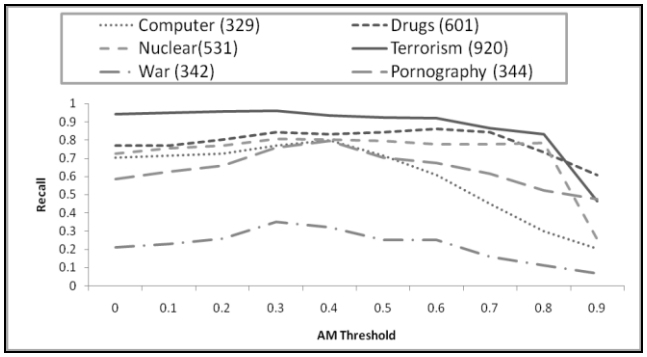


Figure 13. Effects of category distribution in training documents on KDP recall using 20 Newsgroups dataset

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* were related to ongoing wars in Iraq and Afghanistan that in such news articles were associated to *terrorism*. Hence, if there are related categories (such as *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*).

Effects of Ambiguous Passages

Finally, we explore the effects of finding ambiguous passages (passages that belong to more than one category) versus finding unambiguous passages (passages that belong to only one category). During the evaluation phase, each passage was labeled as *ambiguous* or *unambiguous* based on the manual evaluation by two graduate students. For example, consider the passage *"a Web site crack resulting in the e-mail addresses of subscribers to a porn website owners e-mail list"*. This passage is related to both the *Computer Crimes* and *Pornography* categories, and hence, is labeled as *ambiguous*. Presence of such passages may mislead the classifier during the *S-PCP*.

Figure 14 shows that the *KDP* approach performs similarly on both *unambiguous* (58%) and *ambiguous* passages (57.6%) for the *S-PCP* task with respect to the F1 measure. The classification decision is based on high weight terms. For our example, the terms such as *email* and *cracking* point more towards computer crimes. Hence, if a passage contains at least few terms that point to a given category, *KDP* approach forms passages around such keywords. Hence, both unambiguous and ambiguous passages and their categories are correctly predicted.

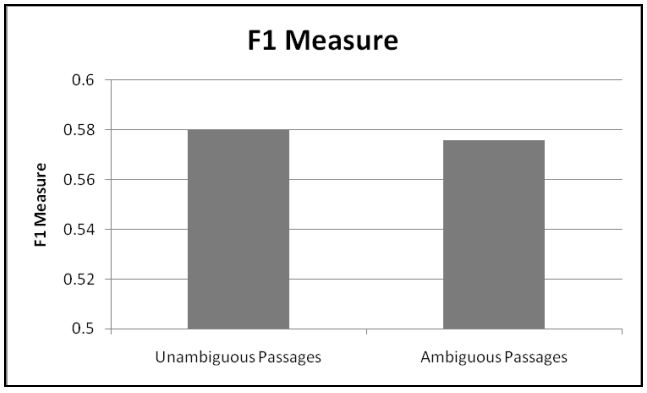


Figure 14. Effects of ambiguous passages on *S-PCP* in KDP approach using 20 Newsgroups

Conclusion

We proposed, designed and evaluated a methodology for detecting passages within documents that belong to the category of user's interest.

We used modified versions of the 20 Newsgroups and Reuters 21578 dataset where passages related to *security* subjects are inserted into selected documents. We simulated the task of detecting such *infected* passages in documents. We evaluated our passage detection approaches for three tasks, namely, *passage detection (PD), stringent passage category prediction (S-PCP)* and *tolerant passage category prediction (T-PCP)*.

We compared the effectiveness of different document splitting techniques and found that *KDP* approach statistically significantly (99% confidence) outperforms *OWP* approach (which is the second best performing document splitting approach) in *PD* (by up to 12%), *S-PCP* (by up to 18%) and *T-PCP* (by up to 12%) with respect to F1 measure. *KDP* approach ensures that each window has at least one keyword. Hence, the precision of all the three tasks increases. As the windows are defined around all the keywords, the recall for detecting both small and large passage is maximized, regardless of the window size. Thus, *KDP* statistically significantly outperforms the other document splitting approaches with respect to F1 measure.

Moreover, applying feature selection statistically significantly (99% confidence) improves the effectiveness of *PD* (by up to 8%), *S-PCP* (by up to 12%) and *T-PCP* (by up to 14%) with respect to the F1 measure for all document splitting approaches.

We also analyzed the effects of different window sizes on passage detection task. Our results indicate that as the size of the window in window passage approaches increases, smaller passages are less detected and larger passages are detected more effectively.

We presented the effects of varying the number of passages in testing documents and showed that as the number of infected passages in testing document increases, the effectiveness of detecting passages in documents also increases.

Our results also indicate that the classification effectiveness for a given category is directly related to the number of training documents available for that category.

References

AbdulJaleel, N., Allan, J., Croft, W. B., Diaz, F., Larkey, L., Li, X., et al. (2004). UMass at TREC 2004: Notebook. *In Text REtrieval Conference.*

Allan, J., Callan, J., Collins-Thompson, K., Croft, B., Feng, F., Fisher, D., et al. (2003). *The lemur toolkit for language modeling and information retrieval.* Retrieved from http://www.cs.cmu.edu/_lemur/.

Callan, J. (1994). Passage Retrieval Evidence in Document Retrieval. *17th ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 302–310).

Goharian, N., & Mengle, S. (2008). On Document Splitting in Passage Detection. *31st ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 833-834).

Grossman, D., & Frieder, O. (2004). *Information Retrieval: Algorithms and Heuristics.* 2nd Edition, Springer Publishers.

Hazel, O. (2002). Email and Internet Monitoring in the Workplace: Information Privacy and Contracting-out. *Industrial Law Journal , Volume 3*, (pp. 321-352).

Hearst, M. (1994). Multi-paragraph segmentation of expository text. *32nd Annual Meeting of the Association for Computational Linguistics*, (pp. 9-16).

Hearst, M., & Plaunt, C. (1993). Subtopic Structuring for Full-length Document Access. *16th ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 59–68).

Hiemstra, D., & Kraaij, W. (1998). Twenty-one at TREC-7: ad-hoc and cross-language track. In E. Voorhees, & D. Harman, *7th Text REtrieval Conference* (pp. 227–238).

Joachims, T. (1998). Text Categorization with Support Vector Machines: Learning with many relevant features. *10th European Conference on Machine Learning*, (pp. 137-142).

Kaszkiel, M., & Zobel, J. (2001). Effective ranking with arbitrary passages. *Journal of the American Society for Information Science and Technology , 52* (4), 344 - 364.

Kaszkiel, M., & Zobel, J. (1997). Passage retrieval Revisited. *20th ACM SIGIR conference on Research and development in information retrieval*, (pp. 178 – 185).

Kim, J., & Kim, M. (2004). An Evaluation of Passage-Based Text Categorization. *Journal of Intelligent Information Systems , 23* (1), 47 – 65.

Lavrenko, V., & Croft, W. B. (2001). Relevance-based language models. *24th ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 120–127).

Ma, L., Goharian, N., Chowdhury, A., & Chung, M. (2003). Extracting Unstructured Data From Template Generated Web Documents. *12th Conference on Information and Knowledge Management*, (pp. 512 - 515).

Mengle, S., & Goharian, N. (2008a). Detecting Hidden Passages from Documents. *8th SIAM Conference on Data Mining Workshop.*

Mengle, S., & Goharian, N. (2008b). Using Ambiguity Measure Feature Selection Algorithm for Support Vector Machine Classifier. *23rd ACM Annual Symposium on Applied Computing*, (pp. 920 – 925).

Mengle, S., Goharian, N., & Platt, A. (2008). Discovering Relationships among Categories using Misclassification Information. *23rd ACM Annual Symposium on Applied Computing*, (pp. 932 – 937).

Miller, D., Leek, T., & Schwartz, R. (1999). A hidden markov model information retrieval system. *22nd ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 214–221).

Mladenić, D., Brank, J., Grobelnik, M., & Milic-Frayling, N. (2004). Feature Selection using Linear Classifier Weights: Interaction with Classification Models. *27th ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 234-241).

Mladenić, D., & Grobelnik, M. (1998). Feature selection for classification based on text hierarchy. *Text and the Web, Conference on Automated Learning and Discovery CONALD-98.*

Nallapati, R. (2004). Discriminative models for information retrieval. *27th ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 64–71).

Ponte, J., & Croft, W. (1998). A language modeling approach to information retrieval. *21st ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 275–281).

Salton, G., Allan, J., & Buckley, C. (1993). Approaches to passage retrieval in full text information systems. *16th ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 49-58).

Salton, G., Allan, J., & Singhal, A. (1996). Automatic text decomposition and structuring. *Information Processing and Management , 32* (2), (pp. 127-138).

Song, F., & Croft, B. (1999). A general language model for information retrieval. *8th international conference on Information and knowledge management*, (pp. 316–321).

Wade, C., & Allan, J. (2005). *Passage Retrieval and Evaluation. CIIR Technical Report*.

Zhou, W., Yu, C., Smalheiser, N., Torvik, V., & Jie, H. (2007). Knowledge-intensive Conceptual Retrieval and Passage Extraction of Biomedical Literature. *30th ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 655 – 662).