Novelty and Redundancy Detection in Adaptive Filtering

Yi Zhang, Jamie Callan
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15232, USA
{yiz,callan}@cs.cmu.edu

Thomas Minka
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15232, USA
minka@cs.cmu.edu

ABSTRACT

This paper addresses the problem of extending an adaptive information filtering system to make decisions about the novelty and redundancy of relevant documents. It argues that relevance and redundance should each be modelled explicitly and separately. A set of five redundancy measures are proposed and evaluated in experiments with and without redundancy thresholds. The experimental results demonstrate that the cosine similarity metric and a redundancy measure based on a mixture of language models are both effective for identifying redundant documents.

1. INTRODUCTION

Information filtering systems monitor document streams to find documents that match information needs specified by user profiles. Most recent research on information filtering focuses on learning to become more accurate at identifying relevant documents, for example, based on long-term observations of the document stream and periodic feedback from the user. This research area is called *adaptive* information filtering, and system performance is typically evaluated using relevancy-based recall, precision, and utility metrics [12].

A common complaint about information filtering systems is that they do not distinguish between documents that contain new relevant information and documents that contain information that is relevant but already known. An information filtering system would provide better service to its users if it identified three categories of documents for each user profile: i) not relevant, ii) relevant but contains no new information, and iii) relevant and contains new information. Users could then decide for themselves how to treat relevant documents that contain no new information.

The decision about whether a document contains new information depends on whether the relevant information in the document is covered by information in documents delivered previously. This complicates the filtering problem. The relevance of a document is traditionally a stateless Boolean value. A document is or is not relevant, without regard to where the document appears in the stream of documents. Decisions about redundancy and novelty depend very much on where in the stream a document appears.

For this study we defined a task and created an evaluation dataset that contains known redundant documents. We model relevance and redundancy separately, and use different similarity measures for relevancy and redundancy. We also developed and tested a variety of redundancy measures.

The following sections describe our efforts towards evaluating and developing algorithms for redundancy/novelty detection while filtering. We begin with a description of the problem and a review of related work. Section 4 describes algorithms for measuring redundancy. Section 5 introduces a simple thresholding algorithm for deciding how much redundancy is "too much". Sections 6 and 7 describe our experimental methodology and results. Section 8 concludes.

2. REDUNDANCY/NOVELTY DETECTION

We want our filtering system to distinguish among relevant documents that contain *new (novel) relevant* information and relevant documents that don't. When a document arrives, the system must determine whether it is on topic (relevancy detection), and if it is on topic, whether it is redundant (redundancy detection). We define "Redundant" to mean that all of the relevant information in the document is covered by relevant documents delivered previously.[1]

The task of identifying novel and redundant documents has not been addressed by prior work, because of the lack of a clear definition of redundancy, and a lack of evaluation data. In the research reported here, novelty and redundancy are defined i) over the set of relevant documents, ii) with respect to previously seen documents, and iii) as opposite endpoints of a scale. The latter point is particularly important. When we treat novelty and redundancy as Boolean values, we imply a thresholding process that maps a value on a continuous redundancy/novelty scale to a Boolean value. We tested our approach to novelty and redundancy by creating an evaluation dataset judged by undergraduate assessors

A system that delivers documents that are novel and relevant must identify documents that are similar to previously delivered relevant documents in the sense of having the same topic, but also dissimilar to the previously delivered documents in the sense of containing new information. These two goals are contradictory, and it may be unrealistic to expect a single component to satisfy them both.

[1]This definition of redundancy includes "duplicate" and "near duplicate" documents as well as documents that are redundant in content but very different in presentation.

SIGIR '02,Tampere, Finland

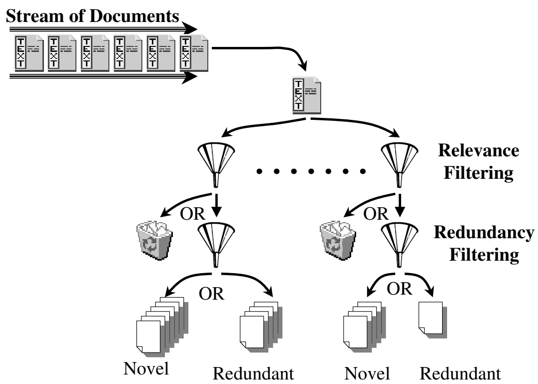

Figure 1: A system that includes traditional document filtering (for relevance) as well as second stage novelty/redundancy detection.

This observation suggests a two stage approach to the problem, as shown in Figure 1. Traditional adaptive information filtering solutions can be used for relevancy filtering. It is less clear what type of algorithm should be used for redundancy filtering, and we defer discussion of solutions to Section 4; for now, we simply observe that a two-stage architecture is likely to simplify the problem.

We use the following notation throughout the paper. All notation is defined with respect to a particular user profile.

- A, B: sets of documents
- d_t: a document that arrives at time t and that is being evaluated for redundancy.
- $D(t)$: the set of all documents delivered for the profile by the time d_t arrives, not including d_t.
- $DR(t)$: the set of all relevant documents delivered for the profile. $DR(t) \subseteq D(t)$.
- $R(d_t)$: the measure of redundancy for document d_t
- d_i: usually refers to a relevant document that was delivered before d_t arrived.

We formulate the task based on the following assumptions, and they are basis for our research when acquiring redundancy judgements and developing algorithms.

Assumption 1 The redundancy of a new document d_t depends on $D(t)$, the documents the user saw before d_t arrived. We use $R(d_t) = R(d_t|D(t))$ to measure this.

Assumption 2 $R(d_t|D(t))$ depends on all the *relevant* documents $DR(t)$ the user has seen when d_t arrives, so $R(d_t|D(t)) = R(d_t|DR(t))$.

Assumption 3 For two documents set A, and B, if $B \subsetneqq A$ and B makes d_t redundant, then A also makes d_t redundant. To make it a softer assumption: $B \subsetneqq A \Rightarrow R(d_t|A) \geqslant R(d_t|B)$

Information filtering systems based on statistical retrieval models usually compute a score indicating how well each document matches a profile; documents with scores above a profile-specific dissemination threshold are delivered. Similarly, the task of identifying redundant documents can be divided into two subtasks: calculate a score to measure how redundant each document is for a profile, then identify documents with scores above a profile-specific redundancy threshold. In our architecture (Figure 1), the second stage redundancy filter consists of two elements: i) redundancy score calculation, and ii) redundancy threshold learning.

In an adaptive filtering system, each of these architectural components defines a different research agenda. The scoring mechanism requires profile-specific "anytime" updating of redundancy measures. The threshold mechanism requires a threshold updating module. The former is the focus of the research described in this paper. Although thresholding is not the focus of this paper, we did implement a simple threshold setting algorithm to make the system more complete, and to enable evaluation of redundancy measures in the context of an operational filtering system.

3. RELATED WORK

The research most closely related to novelty or redundancy detection in adaptive information filtering is perhaps the First Story Detection task associated with Topic Detection and Tracking (TDT) research [1]. A TDT system monitors a stream of chronologically-ordered documents, usually news stories. The First Story Detection (FSD) task is defined as detecting the first story that discusses a previously-unknown event. An *event* is defined as "something that happens at some specific time and place" [14].

Online clustering approaches have been a common solution to the FSD task [10, 3, 2, 5, 4, 13, 15, 1, 14]. New stories are compared to clusters of stories about previously-known events. If the new story matches an existing cluster, it describes a known event, otherwise it describes a new event.

One might argue that the concepts "event" and "novelty" are related, or that solutions defined to detect events also work for novelty. However, we think this unlikely. FSD is an event-based task, and TDT researchers have developed a distinct set of methods for topic tracking. Events have certain structures and occurrence patterns in the media. Information filtering profiles, in contrast, tend to be more subject-oriented and are often intended to follow a subject area over a relatively long period of time. Distinct events might or might not occur in the stream of relevant documents. We would also expect the two tasks to be sensitive to different vocabulary patterns. Indeed, because the user profile and a stream of relevant documents define a far smaller universe of documents than encountered in the TDT task, we might expect novelty/redundancy detection in a filtering environment to be an easier task than FSD in TDT.

FSD is a difficult problem, far from solved. The predicted and actual error rates are unacceptably high for all but a few applications [2, 5]. However, the similarities of the two tasks are worth exploring, and several of the redundancy measures we investigated are motivated by work on FSD.

4. REDUNDANCY MEASURES

We assume that traditional information filtering techniques are used to identify relevant documents; we recognize that the filtering system will make mistakes, i.e., it will deliver some documents that are not relevant and discard some documents that are actually relevant. However, for simplicity we assume that novelty/redundancy detection is performed on a stream of documents that are presumed to be relevant.

We frame this problem as finding a measure $R(d_t|DR(t))$, based on Assumption 2 in Section 2.

One approach to novelty/redundancy detection is to cluster all previously delivered documents, and then to measure the redundancy of the current document by its distance to each cluster. This approach is similar to solutions for the TDT First Story Detection problem. Our concerns about this approach are that it is sensitive to clustering accuracy, and is based on strong assumptions about the nature of redundancy, which we think is user dependant.

Another approach is to measure redundancy based on the distance between the new document and each previously delivered document (document-document distance). This approach may be more robust than clustering, and may be a better match to how users view redundancy. When we asked assessors to annotate an evaluation dataset, we found that it was easiest for them to identify a new document as being redundant with a specific previously seen document, and harder to identify it as being redundant with *a set* of previously seen documents. This observation allows us to simplify the calculation of $R(d_t|DR(t))$ by setting it equal to the value of the maximally similar value in all $R(d_t|d_i)$.

$$R(d_t|DR(t)) = argmax_{d_i \in DR(t)} R(d_t|d_i)$$

Although duplicate detection is not our goal, it is an instructive case because it is simple. If d_t and d_i are exact duplicates ($d_t = d_i$) then $R(d_t|d_i)$ should have a high value because a duplicate document is maximally redundant. One natural way to measure $R(d_t|d_i)$ is using measures of similarity/distance/difference between d_t and d_i.

Document timestamps are also an important source of evidence, because documents are more likely to be redundant with other recently delivered documents. During redundancy decisions truncating the delivery history to the most recent N documents delivered for a profile also reduces the number of documents that must be considered, which reduces computational costs. N is set to 10 in all experiments reported in this paper.

Redundancy is not a symmetric metric. d_j may cause d_k to be viewed as redundant, but if the presentation order is reversed, d_k and d_j may both be viewed as containing novel information. A simple example is a document d_k that is a subset (e.g., a paragraph) of a longer document d_j. This problem characteristic motivates exploration of asymmetric forms of traditional similarity/distance/difference measures.

Below we present several different approaches to redundancy detection. The simple set distance measure is designed for a Boolean, set-oriented document representation. The geometric distance (cosine similarity) measure is a simple metric designed for "bag of words" document representations. Several variations of KL divergence and related smoothing algorithms are more complex metrics designed to measure differences in word distributions.

4.1 Set Difference

The set difference measure represents each document as a set of words. The novelty of a new document d_t is measured by the number of new words in the smoothed set representation of d_t. If a word w_i occurred frequently in document d_t but less frequently in an old document d_i, it is likely that new information not covered by d_i is covered by d_t.

Some words are expected to be frequent in a new document because they tend to be frequent in the corpus, or because they tend to be frequent in all relevant documents. There may also be topic-related stopwords, which are words that behave like stopwords in relevant documents, even if they are not stopwords in the corpus as a whole. An effective measure must compensate for both types of words.

Our set difference measure compensates for corpus stopwords by smoothing a new document's word frequencies with word counts from *all* previously seen documents. It compensates for topic stopwords by smoothing a new document's word frequencies with word counts from *all delivered* (presumed relevant) documents.

Thus we have the following measure for the redundancy of current document d_t with respect to old document d_i.

$$R(d_t|d_i) = \|Set(d_t) \bigcap \overline{Set(d_i)}\| \quad (1)$$

Where:
$w_j \in Set(d)$ iff $Count(w_j, d) > k$
$Count(w_j, d) = \quad \alpha_1 * tf_{w_j,d} + \alpha_2 * df_{w_j} + \alpha_3 * rdf_{w_j}$
$tf_{w_j,d}$: the frequency of word w_j in document d
df_{w_j} : the number of filtered documents that contain w_j
rdf_{w_j} : the number of delivered relevant documents that contain word w_j
(α_1, α_2, α_3, k) are set to (0.8, 0.2, 0.0, 2) in our experiments; they could also be learned from training data.

We are not using the true difference between two sets

$$\|Set(d_t) \bigcap \overline{Set(d_i)}\| + \|\overline{Set(d_t)} \bigcap Set(d_i)\|$$

here because the words in

$$\|\overline{Set(d_t)} \bigcap Set(d_i)\|$$

shouldn't contribute to the novelty of d_t and the optimal novelty measure should be asymmetric.

4.2 Geometric Distance

There are several different geometric distance measure, such as Manhattan distance and Cosine distance [8]. Since Manhattan distance is very sensitive to document length, Cosine distance maybe more appropriate for our task. Prior research showed that a Cosine distance based measure was useful for the TDT FSD task [4].

Cosine distance is a symmetric measure related to the angle between two vectors [6]. If we represent document d as a vector $d = (w_1(d), w_2(d), .., w_n(d))^T$, then:

$$\begin{aligned} R(d_t|d_i) &= \cos d_t, d_i && (2) \\ &= \frac{\sum_{k=1}^{n} w_k(d_t) w_k(d_t)}{\| d_t \| \ \| d_i \|} && (3) \end{aligned}$$

In our study, we used each unique word as one dimension, and set the tf.idf score as the weight of each dimension.

4.3 Distributional Similarity

Probabilistic language models have shown promise for identifying relevant documents in ad-hoc IR tasks (e.g., [10, 7, 16]). In the language model approach, a document d is represented by a unigram word distribution θ_d. Kullback-Leibler divergence, a distributional similarity measure, is one way to measure the redundancy of one document given another.

$$\begin{aligned} R(d_t|d_i) &= -KL(\theta_{d_t}, \theta_{d_i}) && (4) \\ &= -\sum_{w_i} P(w_i|\theta_{d_t}) log(\frac{P(w_i|\theta_{d_i})}{P(w_i|\theta_{d_t})} && (5) \end{aligned}$$

where θ_d is the language model for document d, and is a multinomial distribution.

θ_d can be found by maximum likelihood estimation (MLE):

$$P(w_i|d) = \frac{tf(w_i, d)}{\sum_{w_j} tf(w_j, d)}$$

The problem with using MLE is that if a word never occurs in document d, it will get a zero probability ($P(w_i|d) = 0$). Thus a word in d_t but not in d_i will make $KL(\theta_{d_t}, \theta_{d_i}) = \infty$.

Smoothing techniques are necessary to adjust the maximum likelihood estimation so that the KL-based measure is more appropriate. Prior research shows that retrieval performance is highly sensitive to smoothing parameters. Several smoothing methods have been applied to ad-hoc information retrieval and text classification (e.g., [17, 9]). Based on this prior research, we selected two methods: Bayesian smoothing using Dirichlet priors, and shrinkage.

4.3.1 Bayesian Smoothing Using Dirichlet Priors

This approach to smoothing uses the conjugate prior for a multinomial distribution, which is the Dirichlet distribution [17]. For a Dirichlet distribution with parameters

$$(\lambda p(w_1), \lambda p(w_2), ..., \lambda p(w_n))$$

the posterior distribution using Bayesian analysis for θ_d is

$$P_\lambda(w_i|d) = \frac{tf(w_i, d) + \lambda p(w_i)}{\sum_{w_j}(tf(w_j, d) + \lambda p(w_j))} \quad (6)$$

In our experiments, if w_j is in d_t, we set $\lambda p(w_j) = 0.5$, otherwise $\lambda p(w_j) = 0$.

4.3.2 Smoothing Using Shrinkage

This approach smooths by "shrinking" parameter estimates in sparse data towards the estimates in rich data [9]. This is a special case of the more general Jelinek-Mercer smoothing method, which involves deleted-interpolation estimation of linearly interpolated n-gram models [17]. For estimating the language model of document d, we can shrink its MLE estimator θ_{d_MLE} with the MLE estimator of a language model for general English θ_{E_MLE} and the MLE estimator of a language model for the topic θ_{T_MLE}:

$$\theta_d = \lambda_d \theta_{d\_MLE} + \lambda_T \theta_{T\_MLE} + \lambda_E \theta_{E\_MLE} \quad (7)$$

where $\lambda_d + \lambda_T + \lambda_E = 1$.

λ_E can be estimated from the documents the filtering system has processed, and λ_T can be estimated from the documents the filtering system has delivered (presumed relevant documents). We can derive empirical optimal values for λ_d, λ_T, and λ_E using "leave-one-out" cross validation as described in [9]. In our experiment, we used "leave-0.5-out".

4.4 A Mixture Model

In this section we introduce a new algorithm based on a generative model of document creation. This approach uses probabilistic language models and KL distance as described in Section 4.3. However, this new mixture model measure is based on a novel view of how relevant documents are generated. We can also view it as a language model with a smoothing algorithm designed specifically for our task.

As shown in Figure 2, we assume each relevant document is generated by the mixture of three language models: A General English language model θ_E, a user-specific

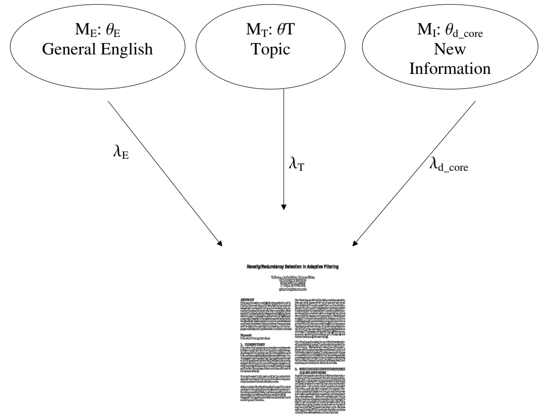

Figure 2: A mixture model for generating relevant documents.

Topic Model θ_T, and a document-specific Information Model θ_{d_core}. Each word w_i in the document is generated by each of the three language models with probability λ_E, λ_T and λ_{d_core} respectively:

$$P(w_i|\theta_E, \theta_T, \theta_{d\_core}, \lambda_E, \lambda_T, \lambda_{d\_core}) = \quad (8)$$
$$\lambda_E P(w_i|\theta_E) + \lambda_T P(w_i|\theta_T) + \lambda_{d\_core} P(w_i|\theta_{d\_core})$$

where $\lambda_E + \lambda_T + \lambda_{d\_core} = 1$.

For example, if information need is "Star Wars", in a relevant document words such as "is" and "the" probably come from the general English model θ_E. Words such as "star" and "wars" probably come from the Topic Model θ_T. For a document with the title "Martin Marietta Is Given Contract For Star Wars Site", the words "Martin" and "Marietta" are more likely to be generated from the new information model θ_{d_core}. θ_T is the core information of a topic, while θ_{d_core} is the core information of a particular relevant document.

$$R(d_t|d_i) \quad = \quad KL(\theta_{d_t\_core}, \theta_{d_i\_core}) \quad (9)$$

If we fix λ_E, λ_T, and λ_{d_core} then there exists a unique optimal value for the document core model $\theta^*_{d\_core}$ that maximizes the likelihood of the document.

$$\theta^*_{d\_core} = argmax_{\theta_{d\_core}} P(d|\theta_E, \theta_T, \theta_{d\_core}, \lambda_E, \lambda_T, \lambda_{d\_core}) \quad (10)$$

Three points are worth noting about the mixture model.

- Although Equations 7 and 8 look similar, the computations performed, and the final model acquired and used to calculate KL divergence, are quite different. Equation 7 uses shrinkage to increase the probability of words that occur frequently in the topic or in general English if they occur less frequently in document d. Equation 8 uses a mixture model to decrease the probability of these words. By shrinking with θ_{E_MLE} and θ_{T_MLE}, shrinkage smoothing reduces the distance between two documents due to those words, thus reducing their effect on the redundancy measure. With the mixture model, we directly decrease the probability of those words to reduce their effect.

- We must fix the values of λ_E, λ_T, and λ_{d_core}. If we train λ's and θ_{d_core} together, we get $\lambda_{d\_core} = 1$ and $\theta_{d\_core} = \theta_{d\_MLE}$, which is the unsmoothed language model for document d; the benefit of smoothing is lost.

- This model intentionally focus on *"what's new"* in a document, thus it avoids the contradiction between identifying relevance and novelty. If the task is to deliver relevant documents, the learning algorithm will try to recognize documents similar to already delivered relevant documents (training data). If the task is to deliver only documents containing novel information, the learning algorithm must avoid documents that are similar to those already delivered. This model introduces θ_T and θ_{d_core}, which means the measure of relevancy and redundancy are focused on different parts of a document. For relevancy, the system would like to focus on θ_T, while for redundancy it should focus on θ_{d_core}. Thus the two tasks are no longer contradictory

We can train $\theta_T, \theta_{d\_score}$, and θ_E using the EM algorithm, which has been used by others to find a language model for similar problems (e.g., [7], [16]).

5. REDUNDANCY THRESHOLDS

When we observed human assessors making redundancy decisions, we found that two annotators working on the same topics sometimes disagreed. Sometimes the disagreement was due to differences in the assessors' internal definition of redundancy. More often one assessor might feel that a document d_t should be considered redundant if a previously seen document d_i covered 80% of d_t; the other assessor might not consider it redundant unless the coverage exceeded 95%.

A person's tolerance for redundancy can be modeled with a user-dependent threshold that converts a redundancy score into a redundancy decision. User feedback about which documents are redundant serves as training data. Over time the system can learn to estimate the probability that a new document with a given redundancy score would be considered redundant : $P(\text{user j thinks } d_t \text{ is redundant}|R(d_t|DR(t)))$.

This two-step process first maps the document d_t into a 1-dimensional space $R(d_t|DR(t)$ using a redundancy measure, and then learns from training data the probability of redundancy given the value on this dimension. This approach is similar in spirit to how many adaptive filtering systems identify relevant documents (e.g., [11, 18]).

Ideally, an optimization goal should be set before deciding what kind of threshold setting algorithm to use. However, our first step is a very simple algorithm for setting thresholds. Our solution is intentionally simple, in part because of the lack of adequate test collection labelled with redundant information for a given profile, and in part because this problem is not (yet) our research focus.

The algorithm for learning redundancy thresholds is:

- Initialize redundancy threshold RThreshold to a value that is so high that only very redundant documents (e.g., near duplicates) are considered redundant; and
- For each document d_t delivered (which means $R(d_t) < RThreshold$ when d_t arrives), ask the user if the document is redundant. If the document is redundant and if $R(d_t) > R(d_i)$ for all $d_i \in DR(t)$
  then $RThreshold = R(d_t)$
  else $RThreshold = RThreshold - \frac{RThreshold - R(d_t)}{10}$.

This is clearly a weak algorithm, because it only decreases the threshold. If the threshold becomes too low there is no method of increasing it again. The effectiveness of this algorithm is explored in Section 7.

6. EXPERIMENTAL METHODOLOGY

6.1 AP News & Wall Street Journal Dataset

We created a one gigabyte dataset by combining AP News and Wall Street Journal data from TREC CDs 1, 2, and 3. We chose these corpora because they are widely available, because information needs and relevance judgements are available from NIST, and because the two newswire corpora cover the same time period (1988 to 1990) and many of the same topics, guaranteeing some redundancy in the document stream. Documents were ordered chronologically. 50 TREC topics (101 to 150) simulated user profiles.

The decision about how to collect redundancy assessments depends in part upon how we view the task. If we viewed redundancy as a relationship between document d_t and a set of documents, for example a subset of the documents delivered for a particular profile, it would be impossible to collect redundancy assessments. We would need to enumerate all of the possible subsets of documents delivered at time t and then ask assessors to judge whether d_t is redundant with respect to each set. The number of possible subsets is 2^{t-1}, which is impractical for all but very small values of t. Although we know that in the "real world" redundancy *is* based on the set of documents delivered previously, we can only model it as a relationship among pairs of documents. This is the approach we adopted when developing algorithms, but that decision was based in part on how we intended to collect redundancy judgements.

We hired undergraduate students, who were otherwise unaffiliated with our research, to read the relevant documents for a profile in chronological order and to provide redundancy judgments. The decision to restrict their attention to relevant documents is based on assumption 2 in Section 2, and is consistent with a filtering system where another component makes decisions about relevance.

Assessors judged one topic at a time. They were instructed to make a decision for each document about whether the information it contained was redundant with document(s) seen previously for that topic, and to identify the prior document(s). Each topic was judged by two assessors and then differences were resolved by the assessors themselves.

We believe that in operational environments different people will have different definitions of redundancy and different redundancy thresholds. We modelled this environment by not giving assessors a precise definition of redundance. We provided two degrees of redundancy, *absolutely redundant* and *somewhat redundant*; assessors could apply them based on their expectations about how a good system should behave. If the assessor thought a person would definitely not want to read d_t because it absolutely contained no new information, d_t was marked as *absolutely redundant*. If the assessor thought that a new document had some new information that a person might want to read, even though much of the document was redundant with a prior document, the document could be marked as *somewhat redundant*. Documents that were not *completely redundant* or *somewhat redundant* were marked as *novel*.

An example of the redundancy assessments is shown below. The first field is a profile id. The second field is the document id of a redundant document. Subsequent document ids are the documents that preceded it in the stream and that made it redundant. A '?' indicates that a document is only partially redundant.

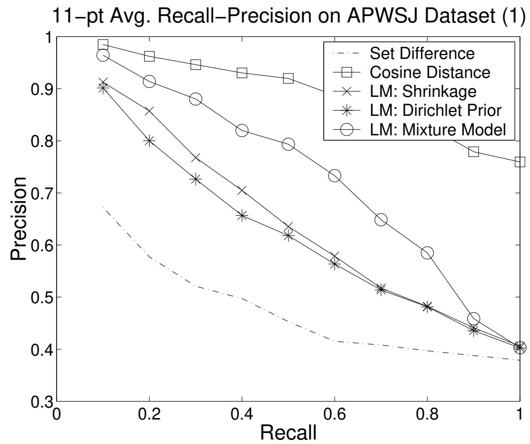

Figure 3: Comparing redundancy measures on AP News & Wall Street Journal data. Documents are considered *redundant* if an assessor marked it *absolutely redundant* or *somewhat redundant*.

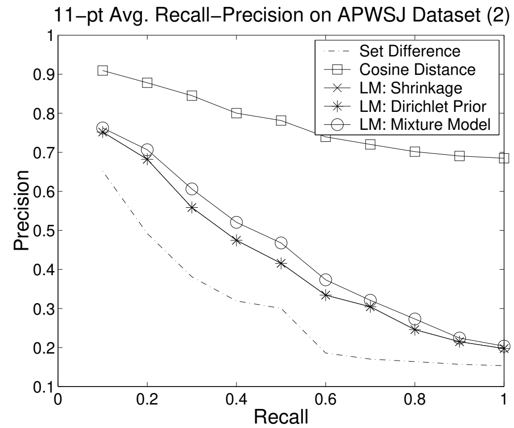

Figure 4: Comparing redundancy measures on AP News & Wall Street Journal data. Documents are considered *redundant* only if an assessor marked it as *absolutely redundant*. The Language Model (LM) measures using Shrinkage and Dirichlet Prior smoothing perform equally, thus overlap.

q121 AP880214-0049 ? AP880214-0002 if user q121 read document AP880214-0002, then AP880214-0049 is somewhat redundant.

q121 AP880217-0031 AP880216-0137 if user q121 read document AP880216-0137, then AP880217-0031 is absolutely redundant.

q128 AP880218-0137 AP880218-0113 AP880218-0112 if user q128 read AP880218-0113 and AP880218-0112, then AP880218-0137 is absolutely redundant.

On average there are about 66.5 records per TREC topic (note that a single record may relate a document to several prior documents). About 19.2 records per profile are *absolutely redundant*; the rest represent partial redundance.

Students reported that the choice of corpus ("old") and topics made this a dull task, so we were unable to collect assessments for all 50 topics. The results in this paper are based on a set of adjudicated assessments for 33 profiles.

6.2 TREC Interactive Dataset

We combined the dataset used by the TREC-6, TREC-7, and TREC-8 Interactive Tracks to create a test dataset containing 210,158 documents from the 1991-1994 Financial Times of London. There are 20 TREC topics; each one defines a user profile. For each topic, TREC assessors have identified several instances. Different instances are about different aspects of the topic. A document on a topic could be mapped to multiple instances of that topic. The mapping from the relevant documents to instances is provided by NIST. In our evaluation, we treated each instance as one aspect of the topic and assumed that a user only wants to see one document on each aspect. Thus a document is redundant if the user has already seen at least one document for each instance this new document belongs to.

Since this dataset was not created explicitly for redundancy detection, it maybe not be as well-matched to the task as the AP News/Wall Street Journal dataset described above. However, we felt that a second dataset, even one that isn't perfect, would be a useful source of information.

6.3 Evaluation Methodology

We believe that it is important to evaluate a particular component of a system with a metric that is not affected by strengths and weaknesses in other parts of system. In this case, we would like to factor out how well the filtering system identifies relevant documents and sets redundancy thresholds. In our experiments we assume that the filtering system identifies relevant documents with 100% precision and recall by evaluating redundancy filtering only on a stream of documents marked relevant by NIST assessors. In some tests we also evaluate the effectiveness of redundancy-scoring algorithms, and factor out the effect of the redundancy threshold algorithm, by reporting average Precision and Recall figures *for redundant documents*. Precision and recall are well-known metrics in IR community. We adapt them to the redundancy detection task as shown below.

$$Redundancy - Precision = \frac{R^-}{R^- + N^-} \quad (11)$$

$$Redundancy - Recall = \frac{R^-}{R^- + R^+} \quad (12)$$

$$Redundancy - Mistake = \frac{R^+ + N^-}{R^+ + N^- + R^- + N^+} \quad (13)$$

R^-, R^+, N^-, N^+ correspond to the number of documents that fall into the following categories:

	Redundant	Non-Redundant
Delivered	R^+	N^+
Not Delivered	R^-	N^-

For simplicity, we will use *Precision* and *Recall* to refer to *Redundancy-Precision* and *Redundancy-Recall* in the rest of this paper.

7. EXPERIMENTAL RESULTS

The five redundancy measures described in Section 4 were compared on the two datasets described in Sections 6.1 and 6.2. A redundancy score was calculated for each relevant

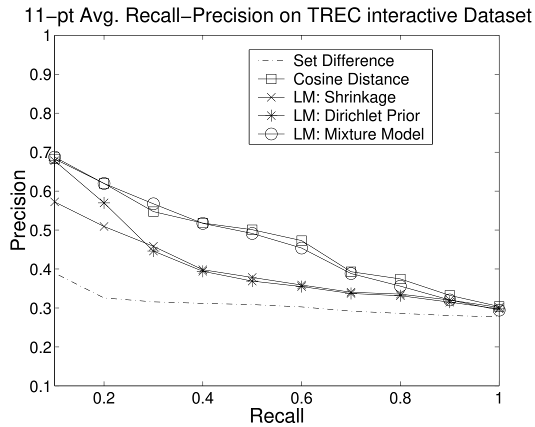

Figure 5: Comparing redundancy measure on TREC Interactive Track data. A document about aspect(s) already covered by previously delivered documents is considered redundant.

document d_t, based on the relevant documents d_i that preceded it in the document stream. The results are shown in Figures 3, 4 and 5 in the form of average Recall-Precision graphs over the set of redundant documents.

On both datasets the Set Difference measure is the least accurate. Representing a document as a set of Boolean word features, even with smoothing to add or delete additional words, was very ineffective.

The traditional cosine-similarity metric (a geometric distance measure) was very effective. This result was a small surprise, because cosine similarity is less well-justified theoretically than the language modelling approaches. The cosine similarity metric is also symmetric; we expected asymmetric measures to be a better model of this task. However, cosine similarity has been demonstrated many times and over many tasks to be a robust similarity metric. Our results add redundancy detection to the long list of tasks for which it is effective.

The results for the three Language Modelling algorithms confirm prior research showing the importance of selecting a good smoothing algorithm. The mixture model approach was consistently more accurate than the other two smoothing algorithms on both corpora. It was also about as effective as the cosine similarity measure on the TREC Interactive Track dataset. This approach to mixing information from corpus, topic, and a document language models provides a new point of view about how to model documents, and it does deliver improved effectiveness compared to other language modelling approaches. This result is not completely surprising, because the algorithm explicitly models *what's new* in a document. However, these results suggest that it is not as robust as the cosine similarity measure.

We also implemented a simple threshold-setting algorithm (Section 5). The threshold-setting algorithm is simple and weak, in part because we did not set an optimization goal specifying the relative rewards and penalties for delivering novel and redundant documents. However, even a simple algorithm can be used to analyze the different redundancy measures. In particular, it provides a more accurate indication of what a user might see in an operational environment.

Tables 1, 2, and 3 summarize the effectiveness of the five redundancy measures when used with the simple redundancy threshold algorithm. Results are reported for both datasets. (The metrics are described in Section 6.3).

Measure	Recall	Precision	Mistake
Set Distance	0.52	0.44	43.5%
Cosine Distance	0.62	0.63	28.1%
LM: Shrinkage	0.80	0.45	44.3%
LM: Dirichlet Prior	0.76	0.47	42.4%
LM: Mixture Model	0.56	0.67	27.4%

Table 1: Average performance of different redundancy measures with a simple thresholding algorithm, measured on 33 topics with the AP News & Wall Street Journal dataset. Both *absolutely redundant* and *somewhat redundant* documents are treated as *redundant.*

Measure	Recall	Precision	Mistake
Set Distance	0.36	0.29	28.1%
Cosine Distance	0.48	0.33	18.7%
LM: Shrinkage	0.375	0.45	21%
LM: Dirichlet Prior	0.375	0.45	21%
LM: Mixture Model	0.46	0.40	16.7%

Table 2: Average performance of different redundancy measures with a simple thresholding algorithm, measured on 33 topics with the AP News & Wall Street Journal dataset. Only *absolutely redundant* documents are treated as *redundant.*

If we evaluate the redundancy measures by the percentage of mistakes they make, the Cosine Similarity and Mixture Model redundancy measures are much better than the rest. These two measures yields a reasonably low percentage of mistakes when a strict definition of redundancy is used (Table 2), but a less satisfying percentage when *somewhat redundant* documents are treated as redundant (Table 1). This result implies that our simple redundancy threshold algorithm models the user by treating *somewhat redundant* as novel. We know that a good threshold setting algorithm is important to system accuracy, and we hypothesize that threshold setting should depends on each user. Table 2 shows that reasonably good accuracy is possible when the thresholding algorithm is well-matched to the task.

8. CONCLUSION

The research reported here is a first step towards adaptive information filtering systems that learn to identify documents that are novel and redundant in addition to relevant and nonrelevant. It defines a task, an evaluation methodology, and a set of novelty/redundancy measures. A reusable corpus was created from generally available documents, a set of adjudicated redundancy judgements was created, and an existing corpus was adapted to our task.

The experimental results demonstrate that it is possible to identify redundant documents with reasonable accuracy. They also demonstrate the importance of a suitable redundancy-threshold algorithm, analogous to the relevance-threshold algorithm found in many information filtering systems. Our results also suggest that the algorithm itself should depend on the user model of redundancy. The extremely small amount of training data (less than what is

Measure	Recall	Precision	Mistake
Set Distance	0.43	0.28	46.8%
Cosine Distance	0.45	0.44	34.5%
LM: Shrinkage	0.79	0.33	53.3%
LM: Dirichlet Prior	0.73	0.34	49.0%
LM: Mixture Model	0.18	0.51	28.4%

Table 3: Average performance of different redundancy measures with a simple thresholding algorithm, measured on 20 topics with the TREC Interactive dataset.

available for relevance-based adaptive filtering) makes it a challenging problem.

We proposed five measures for assessing the redundance of a new document with respect to a previously seen stream of documents. Our experimental results demonstrate that the well-known cosine similarity metric is effective on this new task. They also demonstrate that a new metric based on a mixture of language models can be as effective as the cosine similarity metric in some cases.

We believe that the metric based on a mixture of language models is an important contribution, whether or not it was the most effective algorithm for this task. We believe that viewing documents as a mix of information covered by corpus, topic, and "new information" models is an appropriate model of an information filtering task. The results reported here are a first attempt to apply this approach to a realistic task; we expect to see other attempts in the future.

This research is only a first attempt at redundancy/novelty detection in an adaptive filtering environment, so there are many open problems for future research. Our research on profile-specific "anytime" updating of redundancy measures just scratches the surface. Although cosine similarity worked well in our experiments, we believe that the underlying redundancy relationship is asymmetric, and that asymmetric measures will eventually be more accurate. It is also likely that other features, such as timestamp, document source, phrases, and proper names will be important sources of evidence for novelty decisions.

Our research measured redundancy based on document-document pairs, which is easy to assess and easy to model, but the underlying task is probably better modelled by comparing clusters of delivered documents to the new document. The best choice may be problem-specific, e.g., depending upon corpus or profile characteristics.

9. ACKNOWLEDGMENTS

We thank John Lafferty, Chengxiang Zhai, Yiming Yang, Chun Jin, and Wei Xu for their very helpful discussions and recommendations. This material is based on work supported by Air Force Research Laboratory contract F30602-98-C-0110. Any opinions, findings, conclusions or recommendations expressed in this paper are the authors', and do not necessarily reflect those of the sponsors.

10. REFERENCES

[1] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study. In *Topic Detection and Tracking Workshop Report*. 2001.

[2] J. Allan, V. Lavrenko, and H. Jin. First story detetion in TDT is hard. In *Proc. of the 9th International Conference on Information and Knowledge Management*, 2000.

[3] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proc. of 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998.

[4] J. Carbonell, Y. Yang, R. Brown, C. Jin, and J. Zhang. CMU TDT report 13-14 Nov 2001. In *Topic Detection and Tracking Workshop Report*. 2001.

[5] M. Franz, A. Ittycheriah, J. S. McCarley, and T. Ward. First story detection: Combining similarity and novelty based approaches. In *Topic Detection and Tracking Workshop Report*, 2001.

[6] W. P. Jones and G. W. Furnas. Pictures of relevance. *Journal of the American Society for Information Science*, 1987.

[7] W. Kraaij, R. Pohlmann, and D. Hiemstra. Twenty-one at TREC-8: using language technology for information retrieval. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, 1999.

[8] L. Lee. Measures of distributional similarity. In *Proceedings of the 37th ACL*, 1999.

[9] A. McCallum, R. Rosenfeld, T. Mitchell, and A. Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of The Eighteenth International Conference on Machine Learning*, 1998.

[10] D. R. H. Miller, T. Leek, and R. Schwartz. A hidden markov model information retrieval system. In *Proceedings of the 22th Annual International ACM SIGIR Conferenc eon Research and Development in Information Retrieval*, pages 214–221, 2001.

[11] S. Robertson. Threshold setting in adaptive filtering. *Journal of Documentation*, 2000.

[12] S. Robertson and D. Hull. The TREC-9 Filtering track report. In *The Ninth Text REtrieval Conference (TREC-9)*, 2001.

[13] M. Spitters and W. Kraaij. TNO at TDT2001: Language model-based topic detection. In *Topic Detection and Tracking Workshop Report*. 2001.

[14] N. Stokes and J. Carthy. Combining semantic and syntactic document classifiers to improve first story detection. In *Proceedings of the 24th Annual International ACM SIGIR Conferenc eon Research and Development in Information Retrieval*, 2001.

[15] J. Yamron, S. Knecht, and P. van Mulbregt. Dragon's tracking and detection systems for the TDT2000 evaluation. In *Proceedings of the Broadcast News Transcription and Understanding Workshop*, 1998.

[16] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of Tenth International Conference on Information and Knowledge Management*, 2001.

[17] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proc. of the 24th Annual Int'l ACM SIGIR Conferenc eon Research and Development in Information Retrieval*, pages 334–342, 2001.

[18] Y. Zhang and J. Callan. Maximum likelihood estimation for filteirng thresholds. In *Proc. of the 24th Annual Int'l ACM SIGIR Conferenc eon Research and Development in Information Retrieval*, 2001.