An Intelligent Information Agent for Document Title Classification and Filtering in Document-Intensive Domains

Dawei Song
Knowledge Media Institute,
The Open University
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
d.song@open.ac.uk

Raymond Y.K. Lau
Department of Information Systems,
City University of Hong Kong
Tat Chee Avenue, Kowloon, Hong Kong SAR
raylau@cityu.edu.hk

Peter D. Bruza
School of Information Systems,
Queensland University of Technology
GPO Box 2434, Brisbane, QLD 4001, Australia
p.bruza@qut.edu.au

Kam-Fai Wong
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong,
Shatin, N.T., Hong Kong SAR
kfwong@se.cuhk.edu.hk

Ding-Yi Chen
School of Information Technology and Electrical Engineering
The University of Queensland,
QLD 4072, Australia
dingyi@itee.uq.edu.au

ABSTRACT

Effective decision making is based on accurate and timely information. However, human decision makers are often overwhelmed by the huge amount of electronic data these days. The main contribution of this paper is the development of effective information agents which can autonomously classify and filter incoming electronic data on behalf of their human users. The proposed information agents are innovative because they can quickly classify electronic documents solely based on the short titles of these documents.

Moreover, supervised learning is not required to train the classification models of these agents. Document classification is based on information inference conducted over a high dimensional semantic information space. What is more, a belief revision mechanism continuously maintains a set of user preferred information categories and filter documents with respect to these categories. Preliminary experimental results show that our document classification and filtering mechanism outperforms the Support Vector Machines (SVM) model which is regarded as one of the best performing classifiers.

Keywords:

Information inference, information flow, belief revision, document classification, information agents.

1. Introduction

Effective decision making (e.g., selecting a stock portfolio for investment) is based on accurate and timely information, and in most cases based on a certain volume of timely information. With the rapid growth of the Internet, there has been ever expanding amount of electronic information available. As a result, decision makers often suffer from the so-called problem of information overload [18, 21]. Accordingly, there is a pressing need for the development of intelligent information processing tools to alleviate the information overload problem, and hence to improve the effectiveness of decision making. In situations involving large amounts of incoming electronic information such as defence intelligence, it is increasingly more important for defence analysts to quickly decide whether certain information items should further be scrutinized (either by automatic or manual means) solely based on metadata elements such as *title* descriptions or brief captions. The reason is that it is too time consuming, or too cognitively demanding to process whole documents. Many of us also make such decisions daily while scanning the subject descriptions of emails, or the title captions in the result set from a search engine.

A human agent can quickly make the judgment that a web page title “Welcome to Penguin Books, U.K” refers to Penguin, the publisher. With reference to the text fragment "Linux Online: Why Linus chose a Penguin?", a human agent may readily infer that "Linus" refers to Linus Torvalds, the inventor of Linux system, and the “penguin” mentioned here has to do with the Linux logo. Another text fragment “Antarctic Penguins”, on the other hand, would lead to the judgment that the text is referring to penguins, the birds. If we want to develop intelligent information processing systems to alleviate the problem of information overload, it is essential that these systems are capable of replicating the kind of approximate, associational inference mentioned above. It should be noted that humans generally excel in exercising intelligent information inference given short captions. Confounding information overload is the fact that a decision maker’s interests may change over time due to the evolving nature of the decision making process (e.g., shifting from the interest of identifying suspected terrorist attacks to locating the sources of money laundering). An intelligent information processing system should therefore be able to track a user’s changing interests and maintain an accurate profile of the user’s preferences over time.

This paper proposes an *intelligent information agent* model for document classification and filtering in information intensive application domains. Intelligent information agents are computer programs situated in some environments (e.g., the World Wide Web) for *autonomous* and *adaptive* information retrieval and filtering on behalf of their human users [16]. The proposed agent-based information classification and filtering system comprises two main components – the classification module (i.e., the classification agent) and the belief revision module (i.e., the filtering agent). The former drives the use of information flow model for category learning and information classification based on document titles, and the latter determines which categories of documents are more likely corresponding to a user’s most current information preferences by conducting non-monotonic reasoning [12]. The architecture of our document classification and filtering system is depicted in Figure 1.

INSERT FIGURE 1 HERE

The proposed agent-based intelligent information processing system makes effective judgments what information fragments are "about", or are not "about", even when the fragments are brief or incomplete. The process of making such "aboutness" judgments has been referred to as *information inference* [4, 31]. An information flow inference model has been developed to automatically discover the implicit information flows from the terse text fragments. These flows underpin the approximate, associational inferences mentioned earlier. More specifically, an information inference of *Y* from *X* reflects how strongly *Y* is *informationally contained* within *X*. Information is represented as vectors in a cognitively motivated high dimensional semantic space model, namely Hyperspace Analogue to Language (HAL) [5, 20]. A combination heuristic is developed to render combinations of concepts, for example, noun compounds, into a single vector representation. Information inference can be performed on the HAL spaces via computing *information flows* between vectors or combination vectors. For example, "publisher" is an information flow derived from "Welcome to Penguin Books, U.K". With information inference, the information classification agent can quickly classify the incoming documents into different categories, e.g., "publisher", "birds", "logo", etc. according to short document titles rather than full text. After document classification, the filtering agent will deduce if certain categories of documents are relevant with respect to the user's specific interests. Such a deduction process is underpinned by non-monotonic reasoning. It has been argued that non-monotonic inference plays an important role in IR [37]. For example, beliefs about which search terms are relevant will grow non-monotonically in light of the documents seen during a search process.

At a first glance, the functionality of our intelligent classification and filtering system is similar to that of a Text Categorization (TC) system which assigns a number of pre-defined category labels to documents. A number of statistical learning algorithms such as K-nearest neighbor (KNN) [8], Naïve

Bayes (NB) [23], Support Vector Machines (SVM) [14], and Neural Networks (NN) [22, 23] have been investigated. These classifiers usually classify a document based on its full content and they require the availability of a large number of training documents to achieve a reasonable level of accuracy. Nevertheless, the characteristics of many real-world domains (e.g., defence intelligence) do not match the pre-requisite requirements of the traditional text categorization algorithms. The main contribution of this paper is the illustration of our novel intelligent information agents underpinned by information flow inference and belief revision; these agents specifically aim at autonomous information processing in data intensive domains such as e-mail scanning, Web page browsing, defence intelligence analysis, etc. where:

- Incoming documents need to be classified based on short descriptions (e.g., document titles);
- Labeled training data may not be available;
- A user's preferences (e.g., interests in categories like "publisher", "birds", "logo", etc.) may change upon what the user has seen in the received documents over time.

The rest of the paper is organized as follows. In section 2, an introduction to the information flow model and how it is applied to infer candidate information categories from document titles is illustrated. Section 3 proposes a belief revision mechanism for dynamic user profiling and adaptive information filtering. The effectiveness and efficiency of our intelligent information agent is evaluated in Section 4. Section 5 highlights the related research work for intelligent information agents. Finally, we offer concluding remarks and describe future direction of this work in Section 6.

2. Classifying Document Titles via Information Flow

This section introduces a model of information flow inference and illustrates how it can be applied to derive categories from unlabeled document titles. We begin with a vector representation of information, based on which an information flow computation is performed.

2.1 Knowledge Representation via Hyperspace Analogue to Language

When humans encounter a new concept, they can draw the meaning of the concept via the accumulated experience of the contexts in which the concept appears. This opens the door to "learn" the meaning of a concept through how a concept appears within the context of other concepts. Following this idea, Burgess and Lund developed a representational model of semantic memory called Hyperspace Analogue to Language (HAL), which is a cognitively motivated and validated semantic space model built upon the term co-occurrence relationships derived from a corpus of text [5].

What HAL does is to generate a word-by-word co-occurrence matrix from a large text corpus via a *L*-sized sliding window; all the words occurring within the window are considered as co-occurring with each other. By moving the window across the text corpus, an accumulated co-occurrence matrix for all the words of a vocabulary is produced. The strength of association between two words is inversely proportional to their distance. Given two words, the weight of association between them is computed by $(L - d + 1)$, whereas d is the distance between these words. After traversing the corpus, an accumulated co-occurrence matrix for all the words in a target vocabulary is produced. HAL is direction sensitive, that is, the co-occurrence information preceding and following a particular word is recorded by the corresponding row vector and column vector separately. By way of illustration, the HAL space for the example text "*The effects of spreading pollution on the population of Atlantic salmon*" is depicted in Table 1, assuming a 5-word window (L=5). As an illustration, the term "effects" appears ahead of "spreading" in the window and their distance is 2 words apart. The value of the cell (spreading, effects) is derived by: 5-2+1 = 4. Table 1 shows how the row vectors encode preceding word order and the column vectors encode posterior word order.

INSERT TABLE 1 HERE

The quality of HAL vectors is influenced by the window size; the longer the window, the higher the chance of representing spurious associations between terms. A window size of eight to ten has been used in various studies [4, 5, 30]. Accordingly, a window size of 10 is used in the experiments reported in this paper. Following the tradition of not keeping order information for the co-occurrence based approaches in information retrieval, our approach is to represent a word by a single vector which is the sum of its row and column vectors. It has been demonstrated in the IR literature that the use of HAL in this way produces good performance [5]. Furthermore, it is computationally more cost-effective than keeping the row and column vectors separately. This is particularly important when we are dealing with large collections. However, we would leave it as an open problem for future work to empirically verify whether representing a word as a unified vector gives a better performance than using the row and column vectors separately.

Formally, a concept (term) *c* is a vector: $c = < w_{cp_1}, w_{cp_2}, ..., w_{cp_n} >$ where $p_1, p_2, ..., p_n$ are called the dimensions of *c; n* is the dimensionality of the HAL space, and w_{cp_i} denotes the weight of p_i in the vector representation of *c*. In addition, it is useful to identify the so-called *quality properties* of a HAL-vector. Intuitively, the quality properties of a concept or term *c* are those terms which often appear in the same context as *c*. A dimension is termed a property if its weight is greater than zero. A property p_i of a concept c is termed a *quality property* iff $w_{cp_i} > \partial$, where ∂ is a non-zero threshold value (e.g., the average weight within that vector). To reduce noise in a vector derived from a large corpus, only certain quality properties are kept. Let $QP_{\partial}(c)$ denote the set of quality properties of concept *c* and $QP_{\mu}(c)$ be the set of quality properties above the mean of positive values. $QP(c)$ is short for $QP_0(c)$. HAL vectors are normalized to unit length before information flow computation. The following normalization method has been proposed in our previous work [30] for a dimension p $_j$ in concept c_i:

$$w'_{c_i p_j} = \frac{w_{c_i p_j}}{\sqrt{\sum_k w_{c_i p_k}^2}} \quad (1)$$

For example, the normalized HAL vector for "*spreading"* with reference to the above example is: **spreading** = < the: 0.52, effects: 0.35, of: 0.52, pollution: 0.43, on: 0.35, population: 0.17 >.

In natural language, word compounds often refer to a single underlying concept. As HAL represents words, it is necessary to address the question of how to represent a concept comprising more than one word. Thus we need to address the issue of concept combination, which has been recognized as a remarkable feature of human thinking [10]. From practical point of view, concept combination does not have to be limited to those syntactically valid phrases. A more general and flexible way of concept combination from arbitrary terms should be developed. A simple method is to sum up the vectors of the respective terms to form a compound. However, we apply a more sophisticated concept combination heuristic [4] to the information agent illustrated in this paper. It can be envisaged as a weighted addition of the underlying vectors paralleling the intuition that some terms are more dominant than others in a given concept combination. Dominance is determined by the specificity of a term. For example, "GATT (General Agreement on Tariffs & Trade) Talks" is more related to "GATT*"* than it is about "talks".

In order to deploy the information flow model in an experimental setting, the dominance of a term is determined by its inverse document frequency (*idf*) value. More specifically, terms can be ranked according to its *idf*. Assuming a ranking of terms: $t_1,\ldots,t_m$ (m > 1), terms t_1 and t_2 can be combined to form a compound concept, denoted as $t_1 \oplus t_2$, whereby t_1 dominates t_2 (as it is higher in the ranking). For this combined concept, the degree of dominance is the mean *idf* scores of t_1 and t_2. The process recurses down the ranking resulting in the composed "concept" $((..((t_1 \oplus t_2) \oplus t_3) \oplus \ldots) \oplus t_m)$. If there is only a single term (m =1), the normalized HAL vector is the same as the combination vector.

We will not give a more detailed description of the concept combination heuristic, which can be found in [4]. Its intuition is summarized as follows:

- Quality properties shared by both concepts are emphasized;
- The weights of the properties in the dominant concept are re-scaled higher;
- The resulting vector developed according to the combination heuristic is normalized to smooth out variations due to the different number of contexts the respective concepts appear in.

By way of illustration, we have the following vector "GATT talks", which is a concept combination of the individual HAL vectors for "GATT" and "talks" derived from the Reuters-21578 corpus.

gatt ⊕ talks = < agreement: 0.282, agricultural: 0.106, body: 0.117, china: 0.121, council: 0.109, farm: 0.261, gatt: 0.279, member: 0.108, negotiations: 0.108, round: 0.312, rules: 0.134, talks: 0.360, tariffs: 0.114, trade: 0.432, world: 0.114>

To demonstrate the sensibleness of our concept combination heuristic, we show in the following a fragment of a Reuters article about GATT talks.

GATT OFFICIAL MEETS WITH U.S. FARM LEADERS.

The official in charge of agricultural matters in the new round of global trade talks is in Washington today and tomorrow to meet with congressional and Reagan administration officials. Aart de Zeeuw, chairman of the General Agreement on Tariffs and Trade's negotiating group on agriculture, met this morning with members of the House Agriculture Committee.

2.2 Computing Information Flow

Information inference determines the degree to which a concept c_j can be inferred "informationally" from another concept c_i or combination of a list of concepts $c_{i_1} \oplus \ldots \oplus c_{i_k}$, denoted degree($\oplus c_i \,|\!-\, c_j$). For ease of

exposition, $\oplus c_i$ will be referred to as c_i because compound concepts are also concepts. A HAL vector is used to represent the information "state" [2] of a particular concept or combination of concepts with respect to a given corpus of text. The degree of information flow is directly related to the degree of inclusion between the respective information states represented by HAL vectors. Total inclusion leads to maximum information flow. The following example illustrates the inclusion between "*GATT⊕talks" and "trade"*.

INSERT FIGURE 2 HERE

The degree of inclusion is computed in terms of the ratio of the sum of weights of the intersecting quality properties of c_i and c_j to the sum of the weights of the quality properties of the source c_i.

$$\text{degree}(c_i, c_j) = \frac{\sum_{p_l \in (QP_{\partial_i}(c_i) \cap QP(c_j))} w'_{c_i p_l}}{\sum_{p_k \in QP_{\partial_i}(c_i)} w'_{c_i p_k}} \qquad (2)$$

The above formula expresses that degree is a function with the vector pairs as its domain and the unit interval [0, 1] as its range. The vector c_i is termed the *source* of the information flow and c_j the *target.* The higher the degree of inclusion of the source in the target, the higher the degree of the information flow. Note that all the dimensions in the target vector c_j, i.e., $QP_0(c_j)$, are used. On the other hand, the parameter ∂_i is used to select quality properties in the source vector. The underlying idea of this definition is to make sure that most of the important quality properties of c_i appear in c_j [31]. If ∂_i is too low, there will be much noise leading to many irrelevant inferences (i.e., target vectors containing the noisy dimensions of the source vector). If it is too high, there will not be sufficient information in the source vector for deriving specific information flows. For the experiment depicted below, ∂_i is set to be

greater than the average positive dimensional weight within c_i, i.e., $QP_\mu(c_i)$, which has been reported as the besting performing setting [31]. The following is an example of information flow computation where the weights represent the degree of information containment between GATT $\oplus$ talks and other terms in the vocabulary of the Reuters collection.

INSERT TABLE 2 HERE

2.3 Classification via Information Flow

A HAL space can be produced from a text corpus, which may be dynamic – it could be expanded when new information comes in. The HAL vectors for the concepts embedded in the new data can then be updated accordingly. This module can be pre-processed and kept updated in the background. The HAL space features a module for driving information inference which is sensitive to the context of local data; the local contexts refer to the incoming document titles. Consider the terms $t_{1,} \ldots, t_m$ being drawn from the title of an incoming document *D*. The concept combination heuristic can be used to obtain a combination vector of $t_1 \oplus t_2 \oplus \ldots \oplus t_m$. The information flows from $t_1 \oplus t_2 \oplus \ldots \oplus t_m$ can be calculated. The top M information flows (ranked by association degree) can be taken as the candidate categories of *D*. For the experiments reported below, M is set to be 80; previous experiments in query expansion via information flow shows that employing top 80 information flows produces the best results [4]. This parameter value will be applied to all the experiments reported in this paper. Figure 3 illustrates this methodology.

INSERT FIGURE 3 HERE

By way of illustration, the concept "trade" is inferred from "GATT talks" with a high association degree of 0.96. Accordingly, the document titled "GATT talks" can be classified under the information

category “trade”. Once a set of inferred candidate categories is assigned to a document title, these categories need to be evaluated with respect to the user’s specific preferences in order to decide whether the categorized documents are desirable (i.e., relevant) or not. As a user’s preferences may change over time, maintaining an updated user profile and determining whether a document is relevant or not is an important and challenging task. The next section will describe our belief revision based method to address such a problem.

3. Dynamic User Profiling and Document Filtering

Although the classification module of our intelligent information agent can assign some ‘candidate categories’ to a document, the ultimate goal of the information agent is to deliver relevant documents to its users. Table 3 gives an example of the information categories assigned to ten documents and the final relevance judgments of these documents. The belief revision module determines which document categories are more likely corresponding to a user’s information preferences based on non-monotonic reasoning. The intelligent agent then delivers the documents of those preferred categories to the user. In addition, the belief revision module maintains a set of information categories (i.e., a user profile) which are relevant with respect to the user's specific interests. The challenge here is that the set of categories is only relevant with respect to the user’s interest to a certain degree because the user may not be sure which categories best described their interests. Accordingly, an intelligent information agent should be able to represent and reason about the uncertainty inherent in relevance prediction. A second challenge is that user's interests may change over time, and so the belief revision module is also responsible for revising the agent's beliefs about the user's changing preferences.

INSERT TABLE 3 HERE

3.1 Representing Users' Information Preferences

Conceptually, the list of *possibly* relevant information categories is represented by a set of beliefs in an agent's knowledge base. The changes of these beliefs are characterized by the corresponding *belief functions* under the guiding principle of *minimal* and *consistent* belief changes [1]. An agent's knowledge base is formalized by a belief state (set) *K* which consists of a set of beliefs. In particular, the AGM belief revision framework [1], which is one of the most influential works in the theory of belief revision, underpins the belief revision modules of our intelligent information agents. In this framework, belief revision processes are taken as the transitions among belief states. A belief state *K* is represented by a theory of a classical language *L*. Three principle types of belief state transitions are identified and modeled by the corresponding belief functions: *Expansion* K_α^+, *Contraction* K_α^-, and *Revision* K_α^*. The AGM framework comprises sets of postulates to characterize these functions for consistent and minimal belief revision. In addition, the AGM framework also specifies the constructions of the belief functions based on various mechanisms. One of them is *epistemic entrenchment* (≤) [11]. It captures the notions of *significance*, *firmness*, or *defeasibility* of beliefs. If inconsistency arises after applying changes to a belief set, the least significant beliefs (i.e., beliefs with the lowest entrenchment degree) are given up in order to restore consistency. The belief revision module of our information agent employs the notion of epistemic entrenchment to represent a user's preferences over some information topics (categories), and utilises the AGM belief revision function K_α^* to revise the agent's beliefs about the user's current preferences.

For a computer-based implementation of the AGM belief revision functions, Williams proposed a finite representation of the epistemic entrenchment ordering $B_\leq$ and an iterated belief revision strategy called maxi-adjustment [35, 36]. An agent's belief is represented by a sentence α (e.g., "¬coffee", "trade", etc.) and the associated entrenchment degree *B(α)*. Based on the Maxi-adjustment method, a more efficient Rapid Anytime Maxi-adjustment (RAM) method was developed to support real-world applications [17].

The RAM method is used to develop the belief revision module of the information agents reported in this paper. The details of the RAM algorithm can be found in [17].

3.2 Inducing a User's Category Preferences

A user's interests can be induced based on a set of documents with category labels assigned by the user. In particular, D^+ and D^- denote the set of relevant categories and the set of non-relevant categories respectively. The sets D^+ and D^- can be developed based on the user's direct or indirect relevance feedback (e.g., based on the viewing time of a category of documents). Essentially, three types of categories can be extracted. *Positive categories* represent what topics the user would like to retrieve; *negative categories* indicate the topics that the user does not want; *neutral categories* are not good indicators of user interests.

The following preference induction formula is used to characterise various types of categories and induce the corresponding *preference values*. It is developed based on the Keyword Classifier which was successfully applied to adaptive information filtering [15].

$$preference\,(cat) = \omega \times \tanh\left(\frac{df\,(cat)}{pos} \times \Pr(\mathrm{Re}\,l \mid cat) \times \log_2 \frac{\Pr(\mathrm{Re}\,l \mid cat)}{\Pr(\mathrm{Re}\,l)} - \frac{df\,(cat)}{neg} \times \Pr(Nrel \mid cat) \times \log_2 \frac{\Pr(Nrel \mid cat)}{\Pr(Nrel)}\right) \quad (3)$$

- *cat* is a category descriptor.
- *pos* and *neg* are the learning thresholds for positive and negative categories respectively.
- *tanh* is the hyperbolic tangent function.
- $\omega < 1$ is an adjustment factor for preference induction.
- $Pr(Rel|cat) = df(cat_{rel})/df(cat)$ is the estimated conditional probability that a category is relevant given that it contains the category descriptor *cat*. It is expressed as the fraction of the number of relevant documents which contain the category descriptor *cat* (i.e., $df(cat_{rel})$) over the total number of documents which contain the same category descriptor *cat* (i.e., *df(cat)*).

- *Pr(Nrel|cat) = df(cat_{nrel}) / df(cat)* is the estimated conditional probability that a category is non-relevant if it contains the category descriptor *cat*.
- *Pr(Rel) =* $|D^+| / (|D^+| + |D^-|)$ is the estimated probability that an arbitrary category is relevant.

A positive value of *preference(cat)* indicates that the underlying category descriptor *cat* represents a positive category, whereas a negative preference value implies that *cat* is associated with a negative category. If the absolute preference value of a descriptor is below a threshold λ, the descriptor represents a neutral category. A positive category descriptor is mapped to a positive literal such as ℓ, whereas a negative descriptor is mapped to a negated literal such as $\neg\ell$. The entrenchment degree of a belief is computed according to:

$$B(\alpha_{cat}) = \begin{cases} \dfrac{|\, preference(cat)\,| - \lambda}{1-\lambda} & \text{if } |preference(cat)| > \lambda \\ 0 & otherwise \end{cases} \qquad (4)$$

Table 4 shows the results of applying the aforementioned preference induction method to the sample category descriptors depicted in Table 3. The last column shows the derived entrenchment degrees associated with the beliefs representing some category descriptors. It is assumed that the parameters $|D^+| = |D^-| = 5$, $\beta = 0.95$, $\lambda = 0.3$, pos=5 and neg = 5 are used in the preference induction process. In general, these parameters are estimated based on empirical evaluation against some document collections. As the contents of D^+ and D^- evolve driven by a user's changing preferences, the entrenchment degrees of corresponding beliefs are raised or lowered in the information agent's knowledge base. The changing epistemic entrenchment ordering of beliefs will then generate different non-monotonic consequence relations which underpin the agent's decisions about category relevance at various points of time.

INSERT TABLE 4 HERE

3.3 Reasoning about Relevant Categories

To determine whether a document should be delivered to a user, the belief revision module first needs to infer if the categories assigned to a document title are relevant or not. More precisely, the *degree of relevance* of such a category is computed. If a category is not highly preferred by the user, it is less likely that the corresponding document will be recommended to the user. On the other hand, if a category is a preferred one, its degree of relevance will be used as the basis to compute the relevance scores of the documents. Conceptually, the agent's inference process is characterised by $K \mid\sim \alpha$, whereas K denotes the agent's knowledge base; α is a sentence representing one of the descriptors of the category assigned to a document title, and $\mid\sim$ denotes the non-monotonic inference relation [12].

An entrenchment-based similarity measure $Sim_B(UP,C)$ is developed to approximate the *semantic correspondence* between a user profile *UP* and an information category *C*. The user profile *UP* is represented by the knowledge base K of an information agent. The similarity measure $Sim_B(UP,C)$ is defined by:

$$Sim_B(UP, C) = \frac{\sum_{\ell \in C} [B(\alpha_l) - B(\alpha_{\neg \ell})]}{|N|} \quad (5)$$

The above formula combines the advantages of quantitative ranking and symbolic reasoning in a single framework. The basic idea is that a category *C* is represented by a set of positive literals (category descriptors) $C = \{\ell_1, \ell_2, \ldots, \ell_n\}$. If the agent's knowledge base *K* logically entails an atom ℓ_i, a positive contribution is made to the overall similarity score because of the partial semantic correspondence between *UP* and *C*. On the other hand, if *K* implies the negation of a literal $\ell_i \in C$, it shows the *semantic*

distance between *UP* and *C*. Therefore, the similarity value is reduced by a certain degree. The cardinality of the set *N* defined by $N = \{\ell \in C \mid B(\alpha_\ell) > 0 \vee B(\alpha_{\neg\ell}) > 0\}$ is used to derive the mean similarity value. Given an agent's knowledge base *K = {(¬computer, 0.6), (business, 0.5), (economy, 0.3), (¬internet, 0.2)}*, and three categories C_1 = *{computer, internet}*, C_2 = *{internet, business}*, and C_3 = *{business, economy}*, the category similarity scores are as follows:

$Sim_B(UP, C_1)$ = -0.4; $Sim_B(UP, C_2)$ = 0.15; $Sim_B(UP, C_3)$ = 0.4.

The final retrieval status value *RSV(Doc)* for a document title *Doc* is defined by the product of its information flow score and its category similarity score with cosine normalisation. If the retrieval status value of a title is greater than or equal to a pre-defined threshold, the corresponding document will be recommended to the user by the information agent.

$$RSV(Doc) = \frac{\sum_{C \in Doc} Sim_B(UP, C) \times IF(Doc, C)}{\sqrt{\sum_{C \in Doc} (Sim_B(UP, C))^2} \times \sqrt{\sum_{C \in Doc} (IF(Doc, C))^2}} \quad (6)$$

Assuming that three document titles Doc_1: *((*C_1*,0.3),(*C_3*,0.8))*, Doc_2: *((*C_1*,0.7),(*C_2*,0.5))*, *and* Doc_3: *((*C_2*,0.4),(*C_3*,*0.3)) are assigned to the categories {C_1, C_2, C_3} by the classification module, the retrieval status values of these document titles with respect to the agent's current beliefs *K = {(¬computer, 0.6), (business, 0.5), (economy, 0.3), (¬internet, 0.2)}* regarding the user's preferences can be computed: RSV(Doc_1) = 0.41, RSV(Doc_2) = -0.56, and RSV(Doc_3) = 0.47. In our example, the classification module is relatively certain that document title Doc_2 is about the category C_1 (computer and internet) because a high *information flow score* $IF(Doc_2, C_1)$*=0.7* is derived based on information inference. Nevertheless, the belief revision based user preference prediction module finds that C_1 is less likely to be a user's favorite (e.g., $Sim_B(UP, C_1)$ = -0.4). Therefore, the overall retrieval status value of document title Doc_2 is less than that of the other document titles in our example. If the document delivery threshold of the intelligent information agent is set to 0.4, the agent will recommend document 1 and document 3 to the user and

reject document 2. Such document dispatch decisions have taken into account the user's current information preferences as well as the semantic correspondence between document titles and the information topics (categories).

4. Experiments and Results

The effectiveness of our intelligent information agent is evaluated empirically based on the Modified Lewis ("ModLewis") Split of the Reuters-21578 corpus [19] which is a commonly used benchmark collection for text classification. This corpus consists of 13,625 training documents and 6,188 test documents. The human indexers have identified 135 topics (categories) and labeled the training documents using these topics.

4.1 Experimental Set-up

In our experiments, seventeen categories were selected from among the 135 topics. The training documents were extracted, and the pre-labeled topics were removed (i.e., there is not any explicit topic information in the training set). This training set was then used to construct a HAL space from which information flows could be computed. After removing stop words, the total vocabulary size of the training set was 35,860 terms. In addition, a collection of 1,394 test documents, each of which contained at least one of the 17 selected topics, was formed. Similarly, the topic labels were removed from the test set. With respect to the 17 topics, only 14 of them were assigned to at least one test document. Among these 14 topics, five topics had more than 100 relevant documents and four topics had less than 10 relevant documents. The average number of relevant documents over the 14 topics was 107. We deliberately chose this set of topics because they varied from the most frequently used topics like "acquisition" (we use the real English word "acquisition" instead of the original topic "acq" for information flow computation) to some rarely used topics such as "rye" in the Reuters collection. Table 5 lists the selected topics and their relevance information:

INSERT TABLE 5 HERE

A HAL space was constructed from the training documents using a window size of 10 words (L = 10). Stemming was not performed during the HAL space construction. For each test document title, the title terms were combined into a single title vector using our concept combination heuristic described in Section 2.

4.2 Effectiveness of Information Flow Based Classification (IFC)

The aim of this experiment is to test the effectiveness of deriving categories directly from document titles by employing the information flow method. We only used the titles of the test documents. The average title length was 5.38 words. Concept combination was applied to each document title to build the concept corresponding to title. The top 80 Information flows with associated degrees were derived from each title vector. If a topic appeared in the list of information flows derived from a document title, it was assigned to this title.

4.3 Effectiveness of Hybrid Information Flow Based Classification with Belief Revision (IFC+BR)

This experiment aims to evaluate the contribution of our belief revision module to the overall system effectiveness. We simulated the initial interests of 17 individual users by using the 17 Reuters topics (categories). The top 80 information flows from each category were also included in the simulated user profile. Our evaluation procedure was similar to the one employed by the TREC adaptive filtering task. For instance, after the IFC module assigned the category labels to a document title, these category labels were presented to the belief revision module. The belief revision module then conducted non-monotonic inference (as described in Section 3.3) to predict the *degree* of relevance of these categories with respect to a simulated user profile. If the retrieval status value (RSV) of a document was above a pre-defined threshold, it would be dispatched to the user. After making the document dispatch decision, the belief

revision module of our system could utilize the simulated user relevance feedback [26] to continuously refine a user profile. A relevance feedback file (as employed by the TREC adaptive filtering task) was used to capture simulated user relevance feedback. This file was developed in advance by parsing the category (topic) labels embedded in the Reuters news documents. With the relevance feedback and user profile revision processes, a more accurate representation of the user's interests could be developed over time. As a result, the precision of the proposed intelligent information agent could be enhanced. Given a specific document collection, the arrival sequence of the incoming document titles will not affect the overall precision of our system because the average precision measure does not take the temporal notion into account.

4.4 Comparison with Supervised Classification Approach: Support Vector Machines (SVM)

For a performance comparison of our system with the state-of-the-art supervised learning approaches, we conducted a classification task using Support Vector Machines (SVM), which has been regarded as one of the best performing classifier [38]. SVM follows the structural risk minimization principle from statistical learning theory [14]. The purpose of structural risk minimization is to find a decision function with minimal test error. The decision functions are represented as hyperplanes. Figure 4 shows the intuition of SVM [3]:

INSERT FIGURE 4 HERE

The stars indicate negative instances, and the circles indicate positive instances. The main objective of SVM is to find the decision function D(X) with minimal test error by maximizing the distance between the closest instances to the separating hyperplane (D(X)). In order to obtain the optimal separating hyperplane, one has to solve the following quadratic programming problem by minimizing the following function: $\Phi(W) = (W \bullet W)/2$. Under the constraints of inequality type such as $y_i = [(x_i \bullet W) + b] \geq 1, \quad i = 1,2,...,\ell$, this kind of problem can be solved by Lagrange methods [34].

One of the advantages of SVM is its capability of dealing with the nonlinear separable data set by mapping the data point to another vector space. The following example shows the basic idea of how SVM deals with the nonlinear separable case: there is no way to linearly separate the data set shown in Figure 5. However, if we map the data points in dimensional space of $(x_1 x_2, x_2)$ as shown in Figure 6, we can find a linear separation function for this data set.

INSERT FIGURE 5 HERE

INSERT FIGURE 6 HERE

SVM uses a kernel function to transform data set into another vector space. The kernel function represents the inner product of feature space (i.e., the transformed space). The frequently used kernel functions are polynomial function, Radial basis function (RBF), and sigmoid function. For the SVM implementation, we used Lib SVM [7] with the Radial Basis Function (RBF) kernel. The SVM type was set to nu-SVR in order to obtain a regression function for estimating the relevance between a document and a category. After the training documents were fed to SVM, the relevance between a test document and a category could be computed using the regression function generated by the SVM method. For simplicity, the detailed regression function is not shown here. In our experiment, the document/category labelling information was kept in the training set. Two sets of SVM experiments were conducted. The first experiment used full texts while the second experiment utilized titles only in the test set. Note that the SVM was expected to achieve a better result than our unsupervised approach, because the SVM experiment employed labeled training documents.

4.5 Experimental Results

The major performance measures used in our experiments were Mean Average Precision (MAP) and Average Recall. Precision is the proportion of documents classified with a category that really belong to the category, whereas recall is the proportion of the documents belonging to a category that are actually assigned that category. The MAP is computed across 11 evenly spaced recall points (0, 0.1, 0.2, …, 1.0) for each category and then averaged over all the categories. The average recall is recall averaged over all the categories. In addition, we measured the efficiency of each system by recording the elapsed time of the classification tasks. All the experimental runs were based on the configuration of a single Pentium III 800MHz CPU with 1GB main memory. The experimental results are summarized in Table 6.

INSERT TABLE 6 HERE

4.6 Discussion

First of all, there is no substantial difference among the average recall achieved by all three approaches when classification is conducted based on document titles only. On the other hand, as an unsupervised learning approach, the IFC model performs reasonably well by achieving 77% of the mean average precision produced by the supervised SVM model when both models are applied to document titles only. However, the IFC model is much more efficient than the SVM model. The hybrid IFC and Belief Revision model (IFC+BR) largely improves the mean average precision by +51% when compared with that achieved by the IFC model alone. This indicates that the belief revision mechanism is more precision-oriented rather than recall-oriented. The computational time of the hybrid (IFC+BR) model is longer due to the fact that belief revision is computationally expensive. However, the average time for processing a document (0.08 seconds) by the BR module is still acceptable and it is less than that consumed by the SVM model.

The hybrid (IFC+BR) model demonstrates a comparable effectiveness to the SVM model when full texts are used. Moreover, the hybrid (IFC+BR) model outperforms the SVM model if document titles are used. This result is noteworthy since our approach does not require any manually labeled training data as the SVM does. Furthermore, we have observed that the training process for SVM is computationally expensive (about three days in our experiments). As a consequence, our approach will be more feasible for data intensive domains where manual labeling of training data is almost impossible (even if it is possible for manual labeling, the training time consumed by SVM is prohibitively expensive). As the HAL space has an additive property, it can be built and enriched accumulatively without the need of additional pre-labeled training data.

5. Related Work in Intelligent Information Agents

WebWatcher [13] is an intelligent browsing agent which recommends hyperlinks to a user while the user is browsing the Web. When an agent is initialized, the user is asked to specify their information interests (i.e., a query) in terms of a set of keywords. Feature extraction is conducted by extracting words from a query or annotated hyperlink of a Web page with high TFIDF [28] scores. If there is a significant overlapping between a query and an annotated hyperlink measured in terms of the cosine similarity score of the corresponding TFIDF vectors, the WebWatcher agent will recommend the user to follow that particular hyperlink. One drawback of the WebWatcher agent is that a large number of labeled training Web documents must be available to train the WebWatcher agent before the agent can accurately recommend promising hyperlinks. Our intelligent information agent does not require labeled training documents to construct an effective classifier.

In order to alleviate the problem of information overload, a general architecture of intelligent information agents is proposed [33]. The main functionalities of the intelligent information agents include intelligent search, navigation guide, auto-notification, personal information management, and dynamic personalized Web page retrieval. The underlying document representations and classification modules of the intelligent information

agents are based on the vector space model [28]. For instance, a Web document is represented by a vector of terms weights, and the match of a Web document with respect to a query is measured in terms of their inner product. A user profile consists of a set of categories representing a user's diverse interests; each category is indeed the mean vector computed based on a set of Web documents in which the user is interested. Similarly, Shaw et al. have proposed an agent-based architecture for intelligent information retrieval in a distributed heterogeneous information environment [29]. The prototype system is called AgentRAIDER. Their work mainly focuses on the architectural aspects of distributed intelligent information agents rather than the computational mechanisms which are used to develop such agents [29]. Our work presented in this paper illustrates both the architectural aspects and the computational details for the development of intelligent information agents. More specifically, we concentrate on document classification and dynamic user profiling among the other functionalities of intelligent information agents referred to in [29, 33]. Our agent classification model is based on the novel information flow based inference method [30, 31] rather than the commonly used vector space model. In addition, dynamic user profiling rather than static user profiling is supported by our intelligent information agents.

Fan et. al. have developed a two-stage model for personalized and intelligent information routing of online news [9]. At the first stage, persistent user queries are extracted from rated documents based on Robertson's Selection Value (RSV). At the second stage, genetic programming is applied to discover the optimal ranking function for individual user. Their system outperforms BM25 (Okapi) [27] and support vector machine (SVM) based routing models evaluated based on the TREC-AP and the TREC-Web datasets respectively. Nevertheless, as indicated in their paper [9], one limitation of the two-stage routing model is that supervised training data must be collected from the users. Moreover, a static rather than a dynamic user profile is assumed. Our research work alleviates these two problems.

Chan and Chong have developed an unsupervised classification model for non-textual Web information (e.g., images) retrieval [6]. Different image feature vectors have been constructed and evaluated. The particular unsupervised image classification model is developed based on the Kohonen self-organizing map (SOM). The performance of the SOM classification model is compared with that of the Hierarchical Agglomerative Clustering (HAC) based on hundreds of images. It is found that the SOM classification model can produce effective clusters of semantically related images. Our proposed information classification model is also based on unsupervised learning. However, it is developed according to a novel information inference mechanism and it is applied to textual data rather than image retrieval. Moreover, belief revision logic is employed to support dynamic user profiling in our approach.

Pazzani and Billsus [25] developed a learning information agent called Syskill & Webert which could learn a user profile for the identification of interesting web documents. A separate user profile was created for each individual information topic. Web documents were represented as Boolean feature vectors, and each feature had a binary value indicating if a particular keyword appeared in the document or not. Feature selection was conducted based on Expected Information Gain which tends to select words appearing more frequently in positive documents. The classification mechanism of Syskill & Webert was based on a naïve Bayesian classifier. Supervised learning was employed to train the classifier. Apart from the naïve Bayesian classifier, several other classification approaches such as Nearest Neighbor algorithm, Decision Trees, Rocchio's method, and Neural Networks were also evaluated. Our information flow based classification agents differ from the Syskill & Webert agents in that users rated training documents are not required to train the agents. As a result, our agents are more autonomous and they can operate with minimal human intervention.

Amalthaea is an adaptive multi-agent system for information discovery and filtering [24]. The agent system could learn a user's information preferences by examining the user's browsing history or obtaining direct relevance feedback from the user. Essentially, the vector space model was used to represent a user's information

preferences as well as web documents. Based on genetic algorithms, the Amalthaea agents could evolve to adapt to the user's changing information needs. Moreover, the agents could explore the potential information topics not explicitly requested by the user by means of genetic operators such as mutation and cross-over. However, as described in their paper [24], it might take many generations of agent evolution before a user's information preferences could be learnt. Furthermore, for each generation of agent evolution, users needed to rate an exhaustive list of web documents. This is in fact a quite labor intensive and time consuming process.

6. Conclusions

This paper illustrates the development and evaluation of an agent-based document classification and filtering system. The document classification module of the agent employs an associational inference mechanism driven by information flow computation. In addition, the dynamic user profiling and filtering mechanism is underpinned by the AGM belief revision logic. It is demonstrated that the inferential capability of information flow computation is complementary to the non-monotonic reasoning capability of the belief revision module. By combining these two powerful mechanisms, the proposed intelligent information agents are highly autonomous and efficient since they can accurately classify electronic documents based on document titles only. Experimental results based on the Reuters-21578 corpus show that the proposed IFC classification module demonstrates comparable performance to the SVM model which is regarded as one of the best-performing supervised classification approaches. When the IFC module is combined with the belief revision module, our hybrid approach outperforms the SVM method in terms of both precision and efficiency. Unlike the SVM model, one major advantage of our document classification mechanism is that it does not require any pre-labeled training data. In the future, larger benchmark document collections will be used to further evaluate the effectiveness and efficiency of our intelligent information agents.

References

[1] Alchourrón, C.E., Gärdenfors, P. and Makinson, D. (1985) On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *Journal of Symbolic Logic,* 50, 510-530.

[2] Barwise, J. and Seligman, J. (1997) *Information Flow: The Logic of Distributed Systems.* Cambridge Tracts in Theoretical Computer Science, 44.

[3] Boser, B.E., Guyon, I. and Vapnik, V. (1992) A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory,* 144-152.

[4] Bruza, P.D. and Song, D. (2002) Inferring Query Models by Computing Information Flow. *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM2002),* 260-269.

[5] Burgess, C., Livesay, K. and Lund K. (1998) Explorations in Context Space: Words, Sentences, Discourse. *Discourse Processes, 25(2&3),* 211-257.

[6] Chan, S. and Chong, M. (2004) Unsupervised clustering for nontextual web document classification. *Decision Support Systems*, 37(3), 377-396.

[7] Chang, C. and Lin, C. (2001) LIBSVM : a Library for Support Vector Machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[8] Dasarathy, B.V. (1991) *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques.* IEEE Computer Society Press.

[9] Fan, W., Gordon, M., and Pathak, P. (2006) An integrated two-stage model for intelligent information routing. *Decision Support Systems*, 42(1), 362-374.

[10] Gärdenfors, P. (2000) *Conceptual Spaces: The Geometry of Thought.* MIT Press.

[11] Gärdenfors, P. and Makinson, D. (1988) Revisions of Knowledge Systems Using Epistemic Entrenchment. In *Proceedings of the Second Conference on Theoretical Aspects of Reasoning About Knowledge*, 83-95.

[12] Gärdenfors, P. and Makinson, D. (1994) Nonmonotonic Inference Based On Expectations. *Artificial Intelligence, 65(2)*, 197-245.

[13] Joachims, T., Freitag, D. and Mitchell, T. (1997) Webwatcher: A Tour Guide for the World Wide Web. In *Proceedings of the fifteenth International Joint Conference on Artificial Intelligence*, 770-775.

[14] Joachims, T. (2001) A Statistical Learning Model of Text Classification for Support Vector Machines. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 128-136.

[15] Kindo, T., Yoshida, H., Morimoto, T. and Watanabe, T. (1997) Adaptive Personal Information Filtering System that Organises Personal Profiles Automatically. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, 716-721.

[16] Lau, R. (2002) The State of the Art in Adaptive Information Agents. *International Journal on Artificial Intelligence Tools, 11(1)*, 19-61.

[17] Lau, R. (2003) Context Sensitive Text Mining and Belief Revision for Adaptive Information Retrieval. *Proceedings of the 2nd IEEE/WIC International Conference on Web Intelligence (WI'03)*, October 13 - 17, 2003, Halifax, Canada, 256 - 262, IEEE Press.

[18] Levy, D. (2005) Users and Interaction Track: Memex and Hypertext: To Grow in Wisdom: Vannevar Bush, Information Overload, and the Life of Leisure. *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries,* 281-286.

[19] Lewis, D. (2003), Reuters-21578 corpus, http://www.daviddlewis.com/resources/testcollections/reuters21578/

[20] Lund, K. and Burgess C. (1996) Producing High-dimensional Semantic Spaces from Lexical Co-occurrence. *Behavior Research Methods, Instruments, & Computers, 28(2),* 203-208.

[21] Maes, P. (1994) Agents that Reduce Work and Information Overload. *Communications of the ACM, 37(7),* 31-40.

[22] Menczer, F. and Belew, R. (2000) Adaptive Retrieval Agents: Internalizing Local Context and Scaling UP to the Web. *Machine Learning, 39(2-3)*, 203-242.

[23] Mitchell, T. (1996) *Machine Learning.* McGraw Hill.

[24] Moukas, A. and Maes, P. (1998) Amalthaea: An Evolving Information Filtering and Discovery System for the WWW. *Journal of Autonomous Agents and Multi-Agent Systems, 1(1)*, 59-88.

[25] Pazzani, M. and Billsus, D. (1997) Learning and Revising User Profiles: The identification of interesting web sites. *Machine Learning, 27(3)*, 313-331.

[26] Rocchio, J. (1971) Relevance Feedback in Information Retrieval. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, 313-323.

[27] Robertson, S.E., Walker, S., Sparck-Jones, K., Hancock-Beaulieu, M.M., and Gatford, M. (1995) OKAPI at TREC-3. In *Proceedings of the 3rd Text Retrieval Conference (TREC-3).*

[28] Salton, G. and McGill, M. (1983) *Introduction to Modern Information Retrieval*. McGraw-Hill.

[29] Shaw, N., Mian, A., and Yadav, S. (2002) A comprehensive agent-based architecture for intelligent information retrieval in a distributed heterogeneous environment. *Decision Support Systems*, 32(4), 401-415.

[30] Song, D. and Bruza, P.D. (2001) Discovering Information Flow Using a High Dimensional Conceptual Space. In Proceedings of the *24th Annual International Conference on Research and Development in Information Retrieval (SIGIR'01*), 327-333.

[31] Song, D. and Bruza, P.D. (2003) Towards A Theory of Context Sensitive Informational Inference. *Journal of American Society for Information Science and Technology, 54(4),* 326-339.

[32] Song, D. and Bruza, P.D., Huang, Z., and Lau, R. (2003) Classifying Document Titles Based on Information Inference. *Proceedings of the 14th International Symposium on Methodologies for Intelligent Systems (ISMIS'2003) Conference*. 297-306.

[33] Tu, H-C. and Hsiang, J. (2000) An architecture and category knowledge for intelligent information retrieval agents. *Decision Support Systems*, 28(3), 255-268.

[34] Vapnic, V. (1995) *The Nature of Statistical Learning Theory*. Springer.

[35] Williams, M.A. (1995) Iterated Theory Base Change: A Computational Model. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI'1995)*, 1541-1547.

[36] Williams, M.A. (1997) Anytime Belief Revision. *In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI'1997),* 74-79.

[37] Wong, K.F., Song, D., Bruza, P.D., Cheng, C.H. (2001) Application of Aboutness to Functional Benchmarking in Information Retrieval. *ACM Transactions on Information Systems*, 19(4), 337-370.

[38] Yang, Y. and Liu, X. (1999) A Re-examination of Text Categorization Methods. *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*, 42-49.

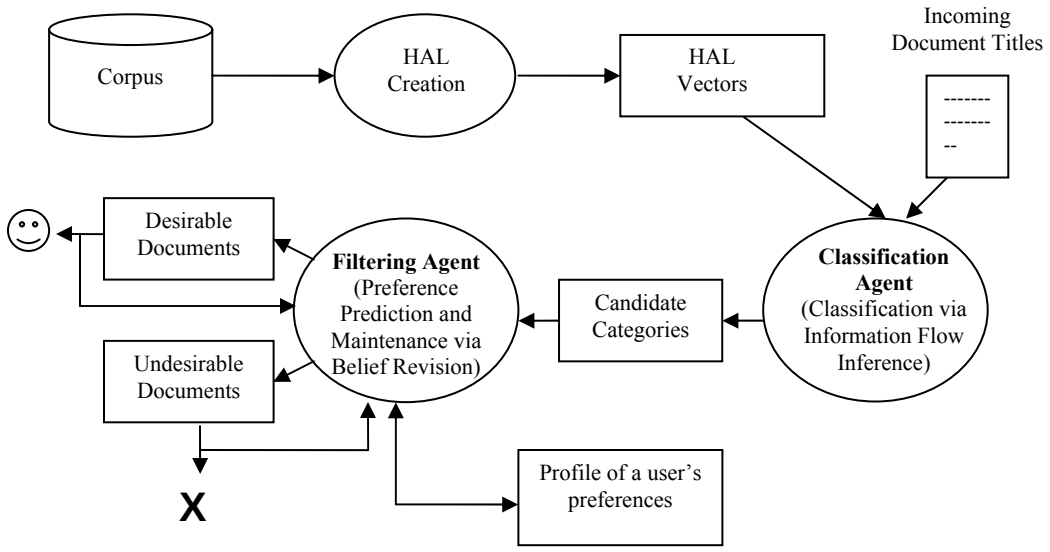

Figure 1: Architecture of the Intelligent Classification and Filtering System

	the	effects	of	spreading	pollution	on	Population	Atlantic	salmon
the		1	2	3	4	5			
effects	5								
of	8	5		1	2	3	5		
spreading	3	4	5						
pollution	2	3	4	5					
on	1	2	3	4	5				
population	5		1	2	3	4			
atlantic	3		5		1	2	4		
salmon	2		4			1	3	5	

Table 1: Example of a HAL Space

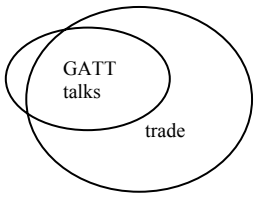

Figure 2: An Example of Inclusion

Degree of information flow	*w*
1.00	GATT
0.96	trade
0.96	agreement
0.86	world
0.85	negotiations
0.84	talks
0.82	set
0.82	states
0.81	EC
0.78	japan

Table 2: Information Flows from "GATT talks"

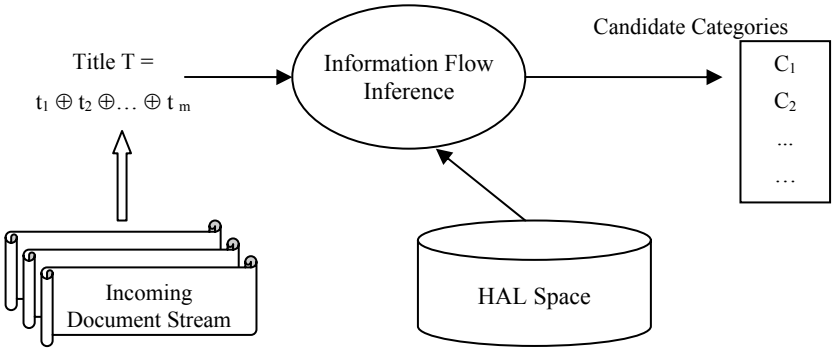

Figure 3: Methodology for Information Flow based Document Title Classification

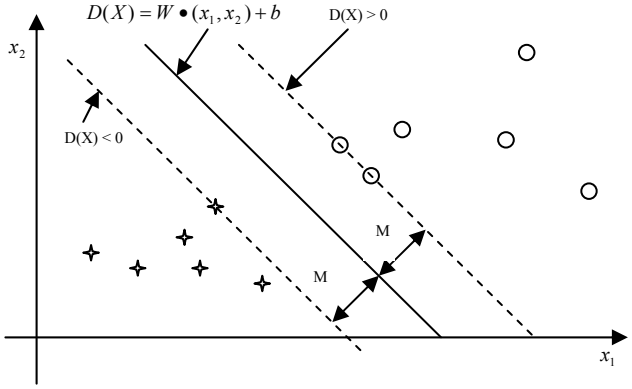

Figure 4: SVM: The objective decision function should be the one that separates the positive instances and negative instances with maximum distance.

Doc1	business	commerce	finance	Relevant
Doc 2	business	commerce	finance	Relevant
Doc 3	business	economy finance	commerce	Relevant
Doc 4	business	economy finance	commerce	Relevant
Doc 5	business	economy finance	commerce	Relevant
Doc 6	finance			Non-Relevant
Doc 7	computer finance	commerce		Non-Relevant
Doc 8	computer finance	commerce		Non-Relevant
Doc 9	computer internet	finance commerce		Non-Relevant
Doc 10	computer internet	finance commerce		Non-Relevant

Table 3: An Example of Category Assignments

Descriptor	**df(cat_{rel})**	**df(cat_{nrel})**	**preference(cat)**	**α_{cat}**	**B(α_{cat})**
business	5	0	0.724	business	0.605
computer	0	4	-0.631	¬computer	0.473
economy	3	0	0.510	economy	0.300
internet	0	2	-0.361	¬internet	0.087
commerce	5	4	0.266	-	-
finance	5	5	0	-	-

Table 4: An Example of Induced Epistemic Entrenchment

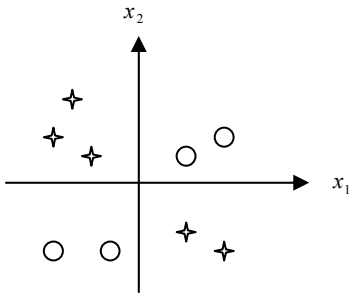

Figure 5: Nonlinear Separable Data Set

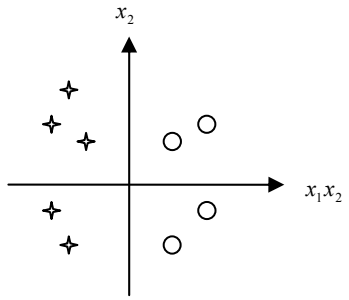

Figure 6: After Transformation

Topics	Number of relevant documents
acq (acquisition)	719
coconut	2
coffee	28
crude	189
grain	149
interest	133
nickel	1
oat	6
peseta	0
plywood	0
rice	24
rupiah	0
rye	1
ship	89
sugar	36
tea	4
trade	118

Table 5: Test Topics (Categories)

Models	Average Precision	Average Recall	Average Elapsed Time (per topic)
IFC (on test document titles)	0.461	0.858	28 seconds
IFC+BR (on test document titles)	0.698	0.861	116 seconds
SVM (on whole test documents)	0.818	0.977	420 seconds
SVM (on test document titles)	0.601	0.878	394 seconds

Table 6: Experimental Results