The Knowledge Engineering Review, Vol. 29:2, 234–247. © Cambridge University Press, 2014
doi:10.1017/S026988891400006X

Information services for novelty mining

FLORA S. TSAI and AGUS T. KWEE

Northwest Indian College, Bellingham, WA 98226, USA;
e-mail: fst1@columbia.edu, atkwee@yahoo.ca

Abstract

Information services facilitate users to exploit applications over the network and access them from the remote system at the client side. In this paper, we describe the design and development of information services for novelty mining, which allows users to access the novel yet relevant information of a given topic. Several methodologies regarding novelty mining such as novelty scoring, novelty threshold, novelty feedback, and document-to-sentence technique are described. In addition to Web services, mobile information services are also described. Modelling and implementing information services for novelty mining are especially useful for users to reduce their information overload. We describe the challenging issue of decomposing the complex novelty mining application into several smaller and simpler modules, which are later implemented as services on the Web as well as mobile devices. After deploying our information services for novelty mining, test cases are provided to demonstrate the system. Our information services for novelty mining are confirmed to be helpful in increasing the efficiency of enterprise users in gathering novel information from incoming text. By studying the design and development of information services for novelty mining, we can benefit other developers in investigating effective techniques for developing enterprise services for other real-world applications.

1 Introduction

With the growth of information technology, the Web is changing from a data-centric Web into a Web of semantic data and services. The demand for Web services that enable users to run software applications over the Internet has rapidly increased. The World Wide Web Consortium defines a Web service as a 'software system designed to support interoperable machine-to-machine interaction over a network' (Hugo & Allan, 2004). More software services are thus available online. The Uniform Resource Indicator serves as the unique address of a particular application or service. Services can be easily created, then combined together to build more complex services. The use of these Web services is important in the business domain, where Web services are used as a means of communicating or exchanging data between businesses and customers. Techniques and technologies for services have evolved rapidly in recent years and now support loose coupling, reusability, composition, discovering, etc. These properties enable better interoperability and information exchange.

Another implication of Web technology is information overload in the form of social networks (Tsai *et al.*, 2009), blogs (Chen *et al.*, 2007), mobile information (Tsai *et al.*, 2010a), etc. Information is abundantly available on the Internet, but much of the information is either irrelevant or repeated. Thus, novelty mining or novelty detection helps solve this problem of information overload (Zhang & Tsai, 2009b). Novelty mining retrieves novel and relevant information from a time-ordered sequence of documents. Novelty mining was first proposed at the document level, where 'novelty' was defined as the opposite of 'redundancy' (Zhang *et al.*, 2002). A document

https://doi.org/10.1017/S026988891400006X Published online by Cambridge University Press

which is less similar to its history documents is considered 'novel' (Liang *et al.*, 2009). Although users can retrieve all the novel documents, they need to read through each document to find the exact novel sentences. Therefore, later studies of novelty mining were performed at the sentence level (Kwee *et al.*, 2009; Ong *et al.*, 2009; Zhang & Tsai, 2009a; Tsai *et al.*, 2010b; Zhang *et al.*, 2011).

Although other technologies such as dimensionality reduction (Tsai, 2010a) and summarisation (Perez-Marin *et al.*, 2009) exist, they have their drawbacks. For example, Google Alerts (Google Inc., 2010) are automatic e-mails or feeds sent when there are new Google results for specific search terms. There are currently six types of alerts—'News', 'Web', 'Blogs', 'Comprehensive', 'Video', and 'Groups' (Google Inc., 2010). However, Google Alerts defines new information based on the time stamp, but does not check for redundancies. For example, a new blog entry can just be a mirror of another blog, but Google Alerts show the new blog entry even if it contains duplicate information. Novelty mining, on the other hand, can eliminate the redundant sentences or documents so that the user only reads the truly novel information.

This paper reports the design and development of information services for novelty mining. Novelty mining services can help enterprise users retrieve new information about events of interest. Using services to facilitate the discovery of novel information is beneficial because it abstracts much tedious user-centric tasks and implement them at the server side (service provider), leaving the user with the simple task of specifying corpus and tuning parameters. The motivation is to offer online services for users to relieve their information burden. Thus, users only need to read novel and relevant information in their topic of interest. This paper significantly advances novelty mining for the benefits of information services. Web services for novelty mining facilitate the rapid deployment and availability of these services, which can greatly benefit the enterprise users.

This paper is organised as follows. Section 2 describes our novelty mining framework. The architecture and design of information services for novelty mining are presented in Sections 3 and 4. Section 5 explains the development of information services on the Web and mobile devices. Two test case scenarios are given in Section 6 to illustrate the information services for novelty mining. Section 7 summarises the entire paper.

2 Detecting novel content

2.1 Overview of novelty mining

The detection of novel content consists of three main steps: (i) preprocessing, (ii) categorisation, and (iii) novelty mining. First, text documents are preprocessed by removing stop words, stemming words to their root forms, etc. Second, each incoming sentence or document (later we only refer to documents without loss of generalisation) is categorised into the relevant topic. Finally, within each topic bin, novelty mining searches through the time sequence of relevant documents and retrieves only those with a minimum amount of novel information.

2.2 Our novelty mining algorithm

Our novelty mining algorithm for the document level can be described as follows. Given a specific topic, all the relevant documents are arranged in a chronological order, i.e. $d_1, d_2, \ldots, d_n$. For each document d_t ($t = 1, \ldots, n$), the degree of novelty of d_t is quantitatively scored by a novelty metric, based on its history documents, that is, d_1 to d_{t-1}. The final decision on whether a document is novel or not depends on whether the novelty score falls above or below a *novelty threshold*. Finally, the current document is pushed into the history document list. This algorithm is also the same for sentence-level novelty mining. The two major components in the algorithm are (i) novelty scoring and (ii) novelty threshold setting. Before we describe the suitable methods for these two components, we will first describe the evaluation measures in the context of novelty mining. This has implications for different types of users with various performance requirements.

https://doi.org/10.1017/S026988891400006X Published online by Cambridge University Press

2.3 Evaluation measures

The system performance can be evaluated using recall and precision. Recall is defined as a fraction of the number of novel documents selected by both the assessor and the system over total number of novel documents selected by the assessor, where precision is defined as a fraction of the number of novel documents selected by both the assessor and the system over total number of novel documents selected by the system. Recall and precision are defined in Equations (1) and (2).

$$\text{Precision} = \frac{M}{S} \tag{1}$$

$$\text{Recall} = \frac{M}{A} \tag{2}$$

where M is the number of novel documents selected by both the assessor and the system; S the number of novel documents selected by the system; and A the number of novel documents selected by the assessor.

There is a trade-off between precision and recall, which can be adjusted using the novelty threshold. A novelty threshold is a cutoff point that determines whether a document is novel. If the novelty score of an incoming document is above this threshold value, this document is predicted as novel. If a user does not know the performance requirements, our system defaults to optimising for F score, which is the primary requirement of previous studies (Soboroff, 2004; Ng *et al.*, 2007). This assumes that the user wants to keep balance between precision and recall. F score is defined in the following equation.

$$\text{F score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{3}$$

In the real world, enterprise users may not be able to judge all the incoming documents. In this case, precision, recall, and F score will be calculated based on a subset of documents.

3 Architecture for novelty mining

3.1 Motivation

In traditional novelty mining, each user has his/her own copy of the novelty mining system. One problem with this model is the update and maintenance of the systems. Furthermore, each user may have different releases of the novelty mining software. This leads to differing results depending on the version used.

However, in the service-oriented version, the novelty mining application resides in the server, which means that novelty mining is performed on the server.

By migrating to a service-oriented Web application, more diverse users will be able to use the services online. Because each user has his/her own requirements for novel information, we need to take the users' preferences into account. First, different users have their own definition of novel information. Some users consider a text is novel when it contains 60% novel information. Some others consider 90% novel information as new information. Second, we need to understand the users' requirements of novel information. For example, if a user does not want to miss any novel information, a high-recall system, which only filters out the most redundant information, is desired. On the other hand, if a user who wants to read the information with the most novelty in a short time, a high-precision system, which only retrieves highly novel information, is preferred.

3.2 Web services protocols

The Simple Object Access Protocol (SOAP) is used for exchanging information over HyperText Transfer Protocol (HTTP) to access a Web service. Because it is interoperable, it can communicate

https://doi.org/10.1017/S026988891400006X Published online by Cambridge University Press

among different platforms and operating systems. For example, a Windows client can send a message to a Unix server for requesting a service. SOAP specifies exactly how to encode the HTTP header and an Extensible Markup Language file so that a client can invoke a program in the server. SOAP also specifies how the program in the server returns the response. Because SOAP can bypass firewall servers, it can communicate with programs anywhere in the server (Duraisamy, 2008). Services need to be engineered in a principled way, and issues of abstraction, autonomy, reusability, discoverability, statelessness, standardised contract, loose coupling, composability—in addition to issues such as availability, scalability, reliability, etc. need to be considered.

The overall process consists of two steps. The first step establishes the protocol between the Web service requester and the provider. The requested Web service is found in the Universal Description, Discovery, and Integration (UDDI) database, which is a directory service to store information about Web services. UDDI allows users or businness to identify themselves by the name, product, location, or service they offer (WhatIs.com, 2003). Web services are not tied to any specific operating system or programming language. UDDI is used to find the location of the requested Web Service Description Language (WSDL) document. WSDL is used to describe and locate a Web service and contains information about the location, the function calls, and the procedure for access. The Web service requesters use the information in the WSDL document to form the SOAP request to the server (James, 2001).

The location of WSDL is used to gather the service information, then the requester performs a SOAP request to invoke the Web service.

The second step exchanges information between the requester and the provider. After a request to a Web service, the server responds by obtaining all the input documents and all the necessary settings. This information uses the SOAP protocol to send the information over the network to the server. The results are processed and returned to the user to be displayed on the client screen.

The Service-Oriented Model (SOA) of Web services is chosen as it allows for looser coupling. SOA relieves the user of the tedious task on mining such data manually or semi-automatically to extract the new information. Furthermore, a lightweight data exchange model is pivotal for adoption, interoperability, and data exchange. Infusing SOA techniques into novelty mining can significantly improve the real-world issues such as information extraction and performance on a network (query latency). Modularising the system, such that the service provider calls other Web services can facilitate looser coupling, such that if a Web service agent (provider) external to the system exists or fails, the system is easily adaptable. These are core issues in SOA with a deployed application and addresses the real challenges of networked Web services interoperability, in addition to deployment and evaluation over a network.

3.3 Service-oriented architecture

The service-oriented architecture contains a total of five sub-applications: *stopWordRem*, *wordStemmer*, *compareTopic*, *compareDocs*, and *predictDocs*. The individual functions implemented as services are described in Table 1.

The services in Table 1 are finally combined to perform the main functions. The categorisation processes: *stopWordRem*, *wordStemmer*, *compareTopic*, and predictDoc services are combined to form *cateProcess*. Likewise, the novelty mining processes: *stopWordRem*, *wordStemmer*, *compareDocs*, and predictDoc are combined into *NMProcess*. Although the novelty mining process contains two levels, document and sentence level, the underlying process is similar. Therefore, only one service is needed for both the document and sentence level, that is, *NMProcess*.

The user first selects either categorisation or novelty mining. The service provider then invokes the required services to perform the process. First, the Parser process, which contains two services, *stopWordRem* and *wordStemmer*, is invoked. The resulting data is then processed by the Metrics process. This process also contains two services, *compareTopic* or *compareDocs* for categorisation or novelty mining, respectively, and *predictDoc*. Finally, the results are returned to the users using the same protocol and displayed on the users' screen.

https://doi.org/10.1017/S026988891400006X Published online by Cambridge University Press

Table 1 Functions of individual services

Service	Function
stopWordRem	Removes stop words of the current document
wordStemmer	Stems the current word into its basic or original form
compareTopic	Compares the current document with the topic information for categorisation
compareDocs	Compares the current document with its history documents for novelty mining
predictDoc	Predicts the relevance or novelty of the current document based on the threshold

4 Design of information services

The main function of each class in the traditional novelty mining system is described as follows.

1. LoadData. This class stores information such as documents and topics into the database. In addition, this class can create a new database.
2. GetData. This class allows users to obtain information such as test documents and relevant documents from the database. The selection of parameters for categorisation and novelty mining processes are provided by this class.
3. API. This class provides the main functions of categorisation and novelty mining at both the document and sentence level.
4. Parser. This class parses the incoming string or text into a collection of words. Stop words removal and word stemming processes are performed. The output of this class is a bag of words that only contains root words (words that have been stemmed).
5. Utility. This class includes auxiliary functions such as calculating a new threshold for automatic threshold setting. Loading dictionary, query, and preparing all queries, are all part of this class.
6. Query. This class is used to create and modify the query table for categorisation.
7. TDMDVector. This class creates a term-document matrix (TDM) of vectors and calculates the norm of a given matrix, which is then compared with the history documents.
8. TDM. This class represents the TDM of a given document.
9. Dictionary. This class manages the process of loading, adding, storing, and deleting terms to and from the database.
10. Metrics. This class stores all the metrics that are available, such as cosine similarity, new word count, and mixed metric.

All the complex applications need to be decomposed into several smaller and simpler applications, which can be implemented as separate services. The resulting class diagram divides the Parser class into two subclasses, *stopWordRem* and *wordStemmer*.

Our approach divides the system according to its functionalities. An important function is novelty scoring, which is a process of assigning a metric score to the current document by comparing it with its previous documents. Many studies have been performed either on creating new metrics or comparing different metrics across different corpora (Zhang *et al*., 2002; Allan *et al*., 2003; Tsai *et al*., 2010b). Because relevance scoring and novelty scoring are similar, these services are combined.

The sentence segmentation (sentSeg) service is used for segmenting a document into its corresponding sentences. By implementing this service, users do not need to manually segment the document into sentences for sentence-level novelty mining. The class diagram after implementing this service is shown in Figure 1.

4.1 Use cases

Two use case scenarios, one for categorisation and another for novelty mining at the document level, are presented in this section.

https://doi.org/10.1017/S026988891400006X Published online by Cambridge University Press

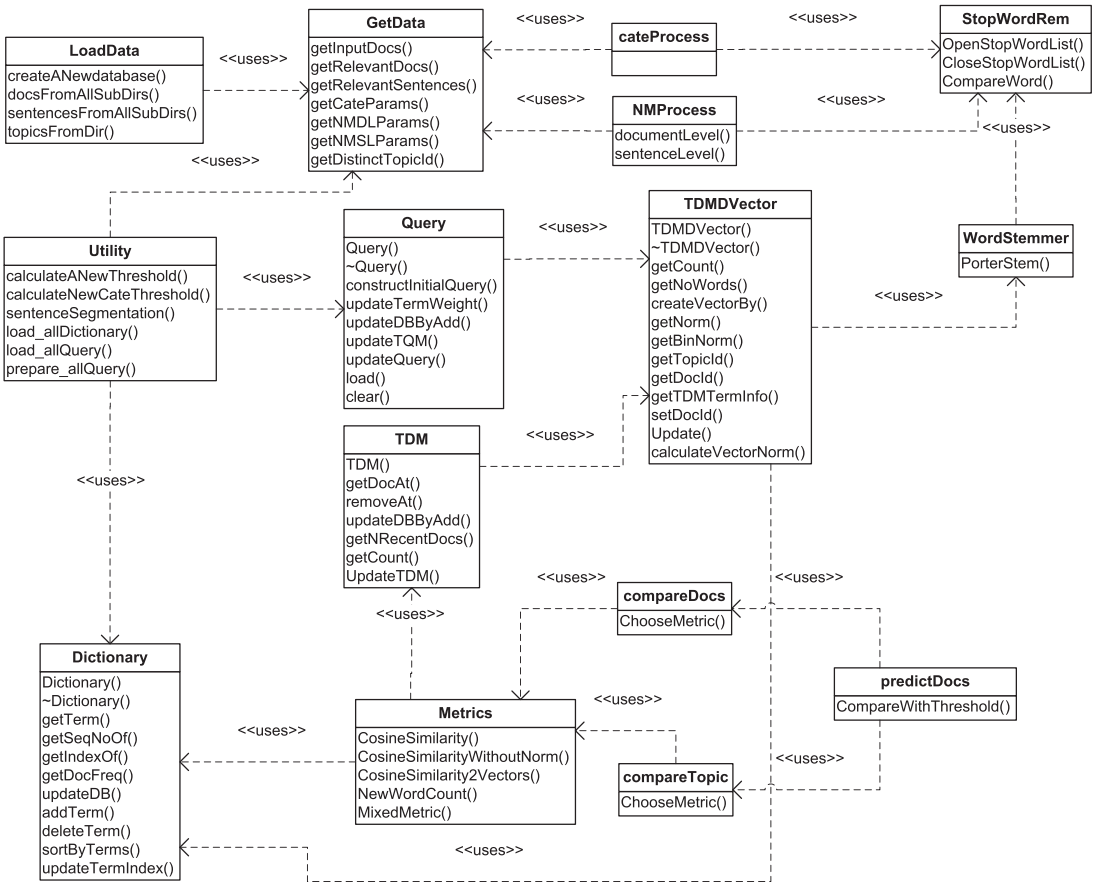


Figure 1 Class diagram of the service-oriented novelty mining system

Actor in the Scenario: *Ann*
Pre-conditions: *Ann has already learnt about the online service-oriented novelty mining system and she also already has a basic knowledge about the system.*

4.1.1 Use case 1

Ann has a set of articles in her hard drive. She wants to find the articles for learning to play the piano. She enters the address of the Web page where the service resides, inputs the topic information and all the input articles. So, she chooses *categorisation* for the process, inputs *Learning Piano* on the topic title field and *How to play piano* on topic description field. She also inputs the articles that may discuss about learning piano. She sets 0.20 for the threshold value. Afterwards, she selects 'Predict', categorisation is performed. Finally, the list of the 50 documents that are retrieved as relevant to the topic (*Learning Piano*) are displayed.

4.1.2 Use case 2

After obtaining 50 articles related to *Learning Piano*, Ann renames and sorts the articles based on the creation date. In order for her not to waste time reading the same information, she uses the service-oriented novelty mining system to find out all novel articles among the 50 articles. Thus, she enters the address of the Web page, then chooses *Document-Level NM* for the desired process. Next, she inputs all 50 documents that have the information about learning piano. She selects a threshold of 0.45, which means that articles with at least 45% novel information will be retrieved. Ann then selects 'Predict' to allow the system to process her query. Upon receiving the data, the server performs novelty mining at the document level. After completing the novelty mining process, the server responds by providing Ann with the list of articles that contain new information about learning piano.

https://doi.org/10.1017/S026988891400006X Published online by Cambridge University Press

5 Development of information services

This section describes the development of information services for novelty mining, which is deployed on both the Web as well as mobile devices. All the input documents and settings are obtained from the user, and all processed data is deleted after the session expires.

We used Visual Basic as the programming language for our services. All five services (*stopWordRem*, *wordStemmer*, *compareTopic*, *compareDocs*, and *predictDoc*) were implemented in Visual Basic.

Three main processes are provided: categorisation, document-level novelty mining process, and sentence-level novelty mining. The user is only allowed to choose one of these options at a given time. For example, the user cannot perform categorisation and document-level novelty mining at the same time.

The user can import text articles by clicking 'Browse'. To actually add the browsed document into the 'Input document': list, 'Add' is selected. 'Remove' is given to remove the unwanted documents in the 'Input documents': list.

The user is also permitted to set the threshold according to his/her own needs. If the user only wants to retrieve the most novel information, he/she can set the threshold higher. On the other hand, if the user does not want to miss any novel information, he/she needs to set the threshold lower. The diverse user definition of 'novelty' can be adjusted by setting the threshold according to the user's requirement. For example, if user A wants to retrieve all documents that contains 75% novel information, he/she sets the threshold to 0.75.

'Predict' is used to send the information to the server to be processed. The results (list of documents along with its scores) are then returned and displayed on the user's screen. 'Clear All' is provided for the user to redo the process without having to reload the page.

5.1 Graphical user interface

In addition to a Web interface, a Graphical User Interface can also be used to call the Web services. The main processes of this application can be implemented as services. They are the categorisation and novelty mining processes at both the document and sentence level.

The system is designed to accommodate different users with various requirements, and can be customised according to an individual user's preference.

The features of the main window are as follows:

1. Select database. This shows the list of existing databases for the current user. The user can also add new databases by selecting add or remove unwanted databases by selecting Remove.
2. Topics. This shows the list of available topics of a database. The user can also add a new topic by selecting Add, modify an existing topic by selecting Edit, and delete unwanted topics by selecting Remove.
3. Test documents. The user can input the test documents by selecting Import. The list box shows the list of documents that have not been categorised or undergone novelty mining, including documents are that are just imported using Import button.
4. Process documents. The user can select the process (categorisation only, novelty mining only, or both). Also, users can select the specific topics.
5. Processed panel. This section is divided into two tabs. One is Processed Docs and the other is Existing Docs. Processed Docs shows the list of the documents being processed during the current session. Existing Docs shows the list of document that were processed on previous sessions. This panel also shows topic id and topic title in the Topic Input box.
6. Viewing panel. This is also divided into two tabs: Sentences and Document View. In the Sentence View, all sentences are listed together with the novelty score and prediction. The Document View displays the content of the selected document.

https://doi.org/10.1017/S026988891400006X Published online by Cambridge University Press

5.2 Categorisation

Users can set properties for categorisation. The properties include:

1. Initial threshold. This threshold is compared with each document's score to determine the relevance of a given document to the specified topic. If the score is above this threshold, the document is predicted to be relevant to the given topic.
2. Alpha, beta, gamma. These parameters are used in the query expansion using Rocchio algorithm as follows:

$$q = \alpha.q_{initial} + \beta.\frac{\sum_{X_i \in R} X_i}{|R|} - \gamma.\frac{\sum_{X_i \in NR} X_i}{|NR|} \quad (4)$$

   where q is the new query updated by introducing training samples and q_{initial} is the initial query learnt from topic information (including title, description, and narratives). R is the set of training samples labelled as relevant and NR is the set of training samples labelled as irrelevant. Parameters α, β, and γ control the weights between initial query, relevant training samples, and non-relevant training samples.
3. Recommended setting. This is the default setting that is proven to be the best according to experiments. Users can choose the desired setting according to their requirements.

5.3 Document-level novelty mining

This option allows users to specify the properties for document-level novelty mining. The properties in this tab contain.

5.3.1 Metrics

Users can select the metric as a basic measurement for detecting novel documents. There are basically two types of novelty metrics, that is, symmetric and asymmetric (Tsai *et al.*, 2010b). Symmetric metric is a metric that yields the same result regardless of the ordering. The most common symmetric metric is the cosine similarity metric, which calculates the similarity between two documents. Then, the novelty score can be obtained by one minus the cosine similarity score.

Another type of novelty metric is the asymmetric metric. This kind of metric takes into account the ordering of documents. A popular asymmetric metric is 'new word count'. This metric counts the number of new words appearing in the current document based on history documents (Allan *et al.*, 2003). This novelty metric was proposed for sentence-level novelty mining (Allan *et al.*, 2003).

As investigated by Tang *et al.* (2010), the strengths of symmetric metrics and asymmetric metrics may complement each other. Therefore, they also proposed a new framework for measuring the novelty by combining both types of novelty metrics. The objective of combining these metrics is to integrate their merits and thus this mixed metric can perform better in the general situation.

For combining strategy, they (Tang *et al.*, 2010) formulated a new framework of measuring through integrating both types of metrics linearly, one of each type, and concluded that this new type of novelty metric shows the superior performance under different performance requirements and for the data with different percentage of novelty ratios. Moreover, this mixed metric is suitable for real-time application for it does not require prior information about the data (Tang *et al.*, 2010).

The three different metrics implemented in our system are:

1. Cosine similarity. This metric compares the similarity between current documents and previous documents in the history set. Documents with lower similarity are considered more novel.
2. New words count. This metric compares the number of the new words between the current documents and previous documents. Documents with more new words are considered more novel.

https://doi.org/10.1017/S026988891400006X Published online by Cambridge University Press

3. Mixed metric. This metric combines the cosine similarity metric and the new word count metric. The weight between these two metrics is controlled by the value α, based on following equation:

$$MMScore = \alpha(1 - CSScore) + (1 - \alpha)NWCScore \quad (5)$$

where $MMScore$, $CSScore$, and $NWCScore$ are the mixed metric score, cosine similarity score, and new word count score, respectively.

5.3.2 *Term weighting function*

This term weighting function (twf) is used only when user chooses the cosine similarity metric. The options are:

1. TF (term frequency). This twf calculates the number of specific term appearing in the document. The larger the number, the heavier the weight of this term.
2. TF-IDF (term frequency—inverse documents frequency). This twf not only calculates the number of specific term appearing in the document, but also calculates the importance of that term across all documents.
3. Binary. This twf only calculates the existence of the word and gives the same weight for each term, regardless of the frequency of the term.

5.3.3 *Additional features*

This feature refers to the additional features that can be chosen. The two features that can be chosen are Named Entity and Term Extractor.

5.3.4 *Threshold*

As stated in Section 2, novelty threshold is used as a boundary value to predict whether an incoming document is novel. Two types of threshold settings are available in our system, that is, fixed threshold and adaptive threshold setting.

As the name suggests, on the fixed threshold setting, the same threshold is used to predict all incoming documents. One question may arise, that is, how to set the right threshold for all the incoming documents. It is nearly impossible to fix a threshold for all documents, since there is little or even no training information in the initial stage of novelty mining.

Therefore, the adaptive threshold setting is introduced as the other alternative in setting the threshold. In this threshold setting, the threshold value is adapted over time as more and more documents come in. We used the adaptive threshold setting algorithm proposed by Tang and Tsai (2009) called Gaussian-based Adaptive Threshold Setting (GATS) because it can not only generalize well on both document-level and sentence-level novelty mining, but also perform robustly on a more practical level where only partial feedback is available (Tang & Tsai, 2009). Another merit of the GATS is that it can satisfy the diverse users' requirements, varying from high-precision to high-recall and also it can suit the diverse users' requirements of 'novelty'. By utilising different optimisation criteria, GATS is able to be tuned according to different performance requirements.

5.3.5 *Process order*

This option represents the sequence of processing documents. Created Date Process Order means that the documents are processed based on the creation date of the documents, whereas Filename Process Order processes the document based on the file name (in descending order).

5.3.6 *History set*

The documents that have already been processed are placed into the history set. This set is compared with the next incoming document. The user can select which documents to be placed inside the history set. All means to place all the already processed documents inside the history set.

https://doi.org/10.1017/S026988891400006X Published online by Cambridge University Press

System Novel means to place the documents that are predicted as novel documents by the novelty mining process.

5.3.7 Document-to-sentence technique

Document-to-Sentence (D2S) is a technique for detecting the novelty of documents using sentence-level information (Tsai & Zhang, 2011). In this technique, sentence-level information was used to predict the novelty of a document, and experimental results showed that the D2S technique outperformed standard document-level novelty mining (Tsai & Zhang, 2010).

To use D2S, two parameters, Novel Rate and Sentence Threshold, need to be provided. Novel Rate is a threshold for deciding whether the documents are novel. The equation for Novel Rate is as follows:

$$\text{Novel Rate} = \frac{\text{number of novel sentences}}{\text{number of sentences}} \tag{6}$$

Sentence Threshold is the threshold used for sentence-level novelty mining, and is similar to the document-level threshold.

5.3.8 Sentence length

This option is used for segmenting documents into sentences. Maximum represents the maximum number of words in a sentence. If the real sentence contains more than the Maximum words, it cuts the sentence and regards these Maximum words as a full sentence. Minimum represents the minimum number of words to be considered as a full sentence. If the number of words in the current sentence is less than Minimum, this sentence is combined with the previous one.

5.3.9 Recommended setting

This is the default setting that is the best setting based on previously conducted experiments.

5.4 Sentence-level novelty mining

These options are similar to the options for document-level novelty mining (Section 5.3), with the addition of heuristic annotation novelty mining (Tsai, 2010b), which uses high-level structures of words, that is, named entities, to accommodate the user's context in the novelty mining system. Four types of entities were given, that is, number, time, location, and person. This system uses a two-layer architecture in detecting novel sentences. The first layer is the original novelty mining algorithm. The second layer uses the heuristic annotation novelty mining algorithm. Every sentence predicted as 'non-novel' in the first layer goes to the second layer. In this layer, the sentence is evaluated based on entities that the user chooses, that is, number/time/location/person.

5.5 Mobile information services

The final task is developing the user interface for the mobile information services. This paper used Netbeans IDE 6.0.1 for the programming language and Sun Java Wireless Toolkit for CLDC 2.5.2 for the simulator. To be able to adapt to the specification of the mobile devices, the user interface has to be modified so that it can read the user input, parse the information, send it to the server, get the response, then parse and display them correctly to the user. Figure 2 shows the user interface of novelty mining process on the mobile device. The sample output is shown in Figure 3. Users can view the document as shown in Figure 4. Using this application, users can retrieve new information regarding a certain topic on a mobile device.

6 Test results

In this section, the results of the two case scenarios (refer to Section 4.1) are presented. One scenario is the categorisation process and the second one is the novelty mining process at the document level.

https://doi.org/10.1017/S026988891400006X Published online by Cambridge University Press

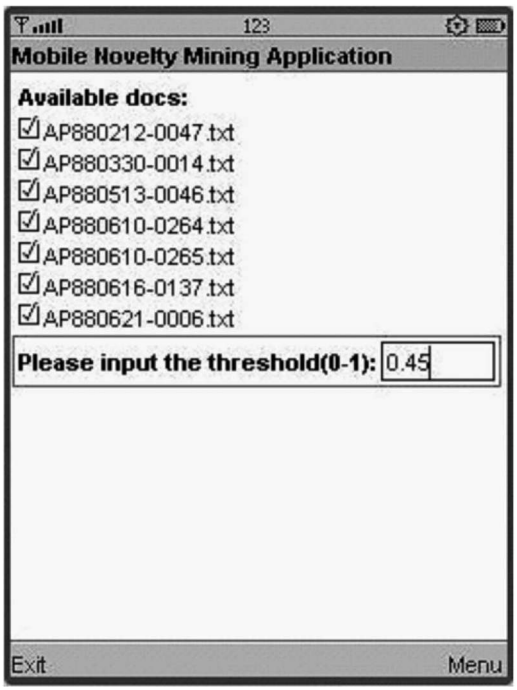


Figure 2 User interface for mobile information services

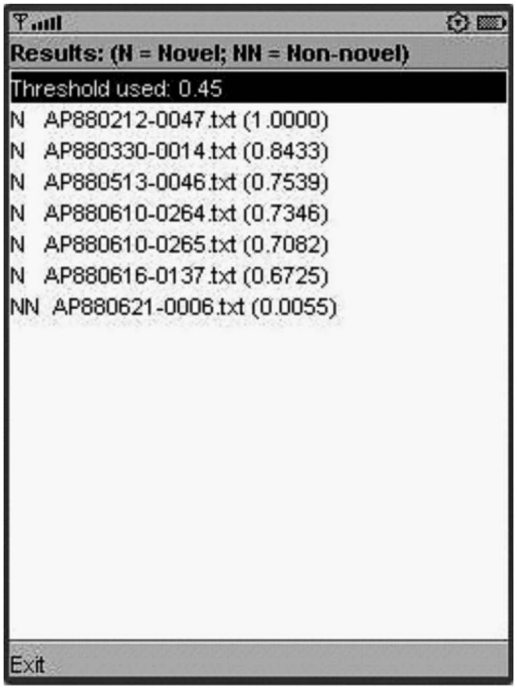


Figure 3 Output for mobile information services

Actor in the Scenario: *Ann*
Pre-conditions: *Ann has already learnt about the online service-oriented novelty mining system and she also already has a basic knowledge about the system.*

https://doi.org/10.1017/S026988891400006X Published online by Cambridge University Press

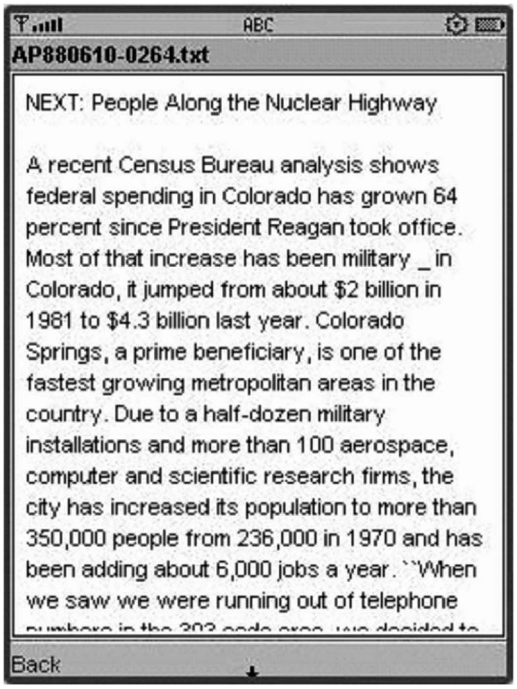


Figure 4 Document view for mobile information services

6.1 Test case 1: categorisation

The result of the categorisation test case gives Ann the list of articles related to learning piano according to the given threshold. It also saves time since she does not need to go through all articles to find the desired ones. Now, let us consider a time critical task. For example, a scientist wants to find all information about swine flu in order for him to find a cure of this disease. If he needs to go through all the documents just to find out the information of this disease, it could take hours or even days. However, by using this service on the categorisation process, he can find all information about swine flu in minutes. Thus, he can create a cure faster and less people would suffer because of this disease.

6.2 Test case 2

The result of the novelty mining test case shows a list of articles that contain new information. The score represents the percentage of novel information in the document. For example, 0.98 means that current document has ~98% of new information compared with its nearest neighbour document. Since these documents are processed chronologically, the user in the financial sector, for example, can make use of this setting to retrieve information about the fluctuation of oil price in the past few weeks, so that he/she can make more accurate decision regarding certain issues.

6.3 Document-level novelty mining results

Our experiments utilised the cosine distance metric with binary weighting, as binary weighting was found to be superior in our previous document-level experiments for data with a high percentage of novel documents. The results for document-level novelty mining are shown in Figure 5 as a graph of Precision–Recall–F score. The grey dashed lines show contours at intervals of 0.1 points of F score. The graph indicates an extremely high precision across most ranges of recall, resulting in the highest F Score of 0.950 at a threshold of 0.25.

https://doi.org/10.1017/S026988891400006X Published online by Cambridge University Press

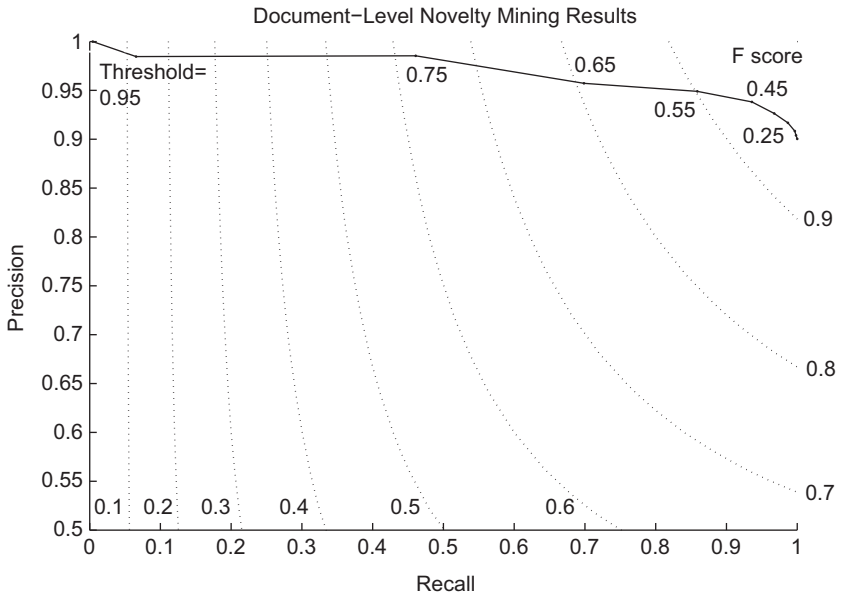


Figure 5 Result of document-level novelty mining

7 Conclusion and future work

This paper describes the modelling and implementation of a service-oriented novelty mining application. We first describe the features of our novelty mining framework, which facilitates users to access the novel and relevant information of a given topic. Then, we presented a service-oriented architecture for novelty mining. By providing novelty mining services, users can access them from anywhere using the Internet. For the design of the service-oriented system, we started from the original novelty mining application, then decomposed it into several smaller and simpler sub-applications, which were subsequently converted into services. These individual services were later combined to perform novelty mining tasks, including categorisation and the detection of novel content. Novelty mining was performed both at the document level and the sentence level. Features of this service-oriented application were provided so that users can use this application effectively. Mobile information services were also deployed for the mobile devices. Two test case scenarios were described to illustrate our newly developed service-oriented novelty mining application. Thus, we have successfully designed, implemented, and deployed the service-oriented novelty mining system that is suitable for enterprise users. By studying the design and development of our novelty mining service application, we can benefit other developers in investigating effective principles and techniques for developing enterprise services for other real-world applications that are able to balance technical concerns with business significance.

Acknowledgments

This research was partially supported by IBM.

References

Allan, J., Wade, C. & Bolivar, A. 2003. Retrieval and novelty detection at the sentence level. In *SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Toronto, Canada, 314–321.

Chen, Y., Tsai, F. S. & Chan, K. L. 2007. Blog search and mining in the business domain. In *DDDM '07: Proceedings of the 2007 International Workshop on Domain Driven Data Mining*, ACM, New York, NY, USA, 55–60.

Duraisamy, S. 2008. *SOAP*. `http://searchsoa.techtarget.com/sDefinition/0,,sid26_gci214295,00.html`

Google Inc. 2010. *Google Alerts*. `http://www.google.com/support/alerts`

https://doi.org/10.1017/S026988891400006X Published online by Cambridge University Press

Hugo, H. & Allan, B. 2004. *Web Services Glossary*. http://www.w3.org/TR/ws-gloss/

James, K. 2001. *Overview of WSDL*. http://developers.sun.com/appserver/reference/techart/overview_wsdl.html.

Kwee, A. T., Tsai, F. S. & Tang, W. 2009. Sentence-level novelty detection in English and Malay. Lecture Notes in Computer Science (LNCS), Springer Berlin/Heidelberg, **5476**, 40–51.

Liang, H., Tsai, F. S. & Kwee, A. T. 2009. Detecting novel business blogs. In *ICICS 2009—Conference Proceedings of the 7th International Conference on Information, Communications and Signal Processing*, Macau, China, 1–5.

Ng, K. W., Tsai, F. S., Chen, L. & Goh, K. C. 2007. Novelty detection for text documents using named entity recognition. In *2007 6th International Conference on Information, Communications and Signal Processing, ICICS*, Singapore, Republic of Singapore, 1–5.

Ong, C. L., Kwee, A. & Tsai, F. 2009. Database optimization for novelty detection. In *ICICS 2009—Conference Proceedings of the 7th International Conference on Information, Communications and Signal Processing*, Macau, China, 1–5.

Perez-Marin, D., Pascual-Nieto, I. & Rodriguez, P. 2009. Computer-assisted assessment of free-text answers. *The Knowledge Engineering Review* **24**(4), 353–374.

Soboroff, I. 2004. Overview of the TREC 2004 Novelty Track. In *Proceedings of TREC 2004—the 13th Text Retrieval Conference*, Gaithersburg, Maryland, USA, 1–16.

Tang, W. & Tsai, F. S. 2009. Threshold setting and performance monitoring for novel text mining. In *Society for Industrial and Applied Mathematics—9th SIAM International Conference on Data Mining, Proceedings in Applied Mathematics 3*, Sparks, Nevada, USA, 1310–1319.

Tang, W., Tsai, F. S. & Chen, L. 2010. Blended metrics for novel sentence mining. *Expert Systems with Applications* **37**(7), 5172–5177.

Tsai, F. S. 2010a. Comparative study of dimensionality reduction techniques for data visualization. *Journal of Artificial Intelligence* **3**(3), 119–134.

Tsai, F. S. 2010b. Review of techniques for intelligent novelty mining. *Information Technology Journal* **9**(6), 1255–1261.

Tsai, F. S., Etoh, M., Xie, X., Lee, W.-C. & Yang, Q. 2010a. Introduction to mobile information retrieval. *IEEE Intelligent Systems* **25**(1), 11–15.

Tsai, F. S., Han, W., Xu, J. & Chua, H. C. 2009. Design and development of a mobile peer-to-peer social networking application. *Expert Systems with Applications* **36**(8), 11077–11087.

Tsai, F. S., Tang, W. & Chan, K. L. 2010b. Evaluation of metrics for sentence-level novelty mining. *Information Sciences* **180**(12), 2359–2374.

Tsai, F. S. & Zhang, Y. 2011. D2S: document-to-sentence framework for novelty detection. *Knowledge and Information Systems* **29**(2), 419–433.

WhatIs.com. 2003. *UDDI*. http://searhsoa.techtarget.com/sDefinition/0,,sid26_gci508228,00.html

Zhang, Y., Callan, J. & Minka, T. 2002. Novelty and redundancy detection in adaptive filtering. In *SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, 81–88.

Zhang, Y. & Tsai, F. S. 2009a. Combining named entities and tags for novel sentence detection. In *Proceedings of the WSDM'2009 ACM Workshop on Exploiting Semantic Annotations in Information Retrieval, ESAIR 2009*, Barcelona, Spain, 30–34.

Zhang, Y. & Tsai, F. S. 2009b. Chinese novelty mining. In *EMNLP 2009: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Singapore, Republic of Singapore, 1561–1570.

Zhang, Y., Tsai, F. S. & Kwee, A. T. 2011. Multilingual sentence categorization and novelty mining. *Information Processing and Management: An International Journal* **47**(5), 667–675.

https://doi.org/10.1017/S026988891400006X Published online by Cambridge University Press