ONTOLOGY-BASED CLASSIFICATION OF NEWS IN AN ELECTRONIC NEWSPAPER

Lena Tenenboim, Bracha Shapira, Peretz Shoval

Abstract: This paper deals with the classification of news items in ePaper, a prototype system of a future personalized newspaper service on a mobile reading device. The ePaper system aggregates news items from various news providers and delivers to each subscribed user (reader) a personalized electronic newspaper, utilizing content-based and collaborative filtering methods. The ePaper can also provide users "standard" (i.e., not personalized) editions of selected newspapers, as well as browsing capabilities in the repository of news items. This paper concentrates on the automatic classification of incoming news using hierarchical news ontology. Based on this classification on one hand, and on the users' profiles on the other hand, the personalization engine of the system is able to provide a personalized paper to each user onto her mobile reading device.

Keywords: Classification, Electronic newspaper, Information filtering, Mobile device, Ontology, Personalization.

ACM Classification Keywords: H5.2. Information interfaces and presentation, H.3.3. Information Storage and Retrieval; information filtering, H.3.1. Content Analysis and Indexing

Conference: The paper is selected from International Conference "Intelligent Information and Engineering Systems" INFOS 2008, Varna, Bulgaria, June-July 2008

1. Introduction

Electronic, online newspapers started appearing at about the same time that the Internet became public. An electronic newspaper has many forms. One form is electronic edition of the printed newspaper (namely, the publisher publishes its "standard" newspaper on a Website, using e.g. PDF files). The user can read the electronic edition similar to a paper edition; there is no personalization, neither with respect to content nor with respect to layout. Another form of electronic newspaper is news website, which enables the user browsing in menus that are organized in subject categories and sub-categories. Yet, another form of electronic newspaper can be seen more like a search engine, which enables the user to insert search terms (i.e., topics of interest) and get in response respective news items that are published on the Web by various news providers. In contrast to the previous forms of services, this one does not publish and does not edit news; rather, it searches and provides links to news published elsewhere by news providers/agencies. In addition to searching according to user-defined terms, such systems enable personalization: the user can define a profile by selecting topics of interest, and the system would search for items in the selected topics. Personalization features are supported in some sites via RSS technology, like in portals that deliver news and have built-in news aggregator capabilities. Personalization is sometimes done based on collaborative filtering methods (e.g. Google News). Interactivity is an additional feature, and exists in the form of reader feedback capabilities and in the form of capabilities for "pulling" personalized news and other information. The general picture is that of much diversity and heterogeneity between the various players in the online newspaper branch of the newspapers industry.

Common to most of the above forms of electronic newspapers is that the user is assumed to read the news from a computer screen, while connected via the Internet to a certain news provider or search engine. But these services might not be sufficient for many readers and reading situations. A newspaper reader may be willing to read articles from various favorite daily and weekly newspapers and magazines while being on the move; for example, while on a business trip, or on vacation, or waiting for a fried at a café, etc. A reader would prefer that someone or something would make interesting articles, from favorite sources; accessible to her all the time and anywhere, delivered directly onto a mobile reading device. A reader would like to subscribe to one unique, personalized newspaper that includes the interesting articles from favourite sources, arranged and presented in an order that best fits her interests and reading habits. An advanced electronic newspaper service should enable a reader, with just one click of a button, to check whether updates of her personal newspaper are available, or to choose receiving updates automatically as soon as they are published.

This paper deals with such a service. *ePaper* is a prototype of an electronic newspaper system aims at providing personalized newspaper on mobile reading devices. The ePaper is projected to provide a "look and feel" of a newspaper that is run on a medium-format mobile device, providing up-to-date news aggregated from many news providers, and personalized according to each user's preferences. The *ePaper* system is a client-server application: On the server side, it aggregates news coming continuously from many news providers; classifies each news item to subject concepts, based on a news ontology; and then determines the relevancy of the news items to each of the subscribed users (readers), based on both content-based and collaborative filters, and ranks the news items that will be delivered to each user, thus providing personalized newspapers. On the client side, the user, who gets the news on her reading device, enjoys an intuitive interface, enabling easy navigation and browsing, and advanced content adaptation capabilities, including switching and configuring layouts. This paper concentrates on only one part of the system - the classification of news items that are obtained by the system, so that later on the personalization algorithms can determine the relevancy of each item to each user.

The rest of this paper is structured as follows: Section 2 presents the general architecture of the ePaper system; Section 3 describes the content managed layer, which is, among else, in charge of the classification of news. Section 4 surveys some related work on classification methods, and Section 5 describes the classifier implemented in the ePaper. Section 6 summarizes and discusses further research.

2. General Architecture of ePaper System

The ePaper is a research project aimed at developing a prototype system that can be viewed as a central newspapers or magazines provider. On one hand, it obtains news obtained from various providers; on the other hand, it distributes personalized newspapers to subscribed users on specialized mobile reading devices. In this section, we describe briefly the general architecture of the system. Figure 1 presents an overview of the ePaper architecture.

The system is implemented based on client-server architecture. The server system consists of five layers: *Aggregator, Content Manager, Personalization, Content Delivery Services,* and *System Management Tools*.

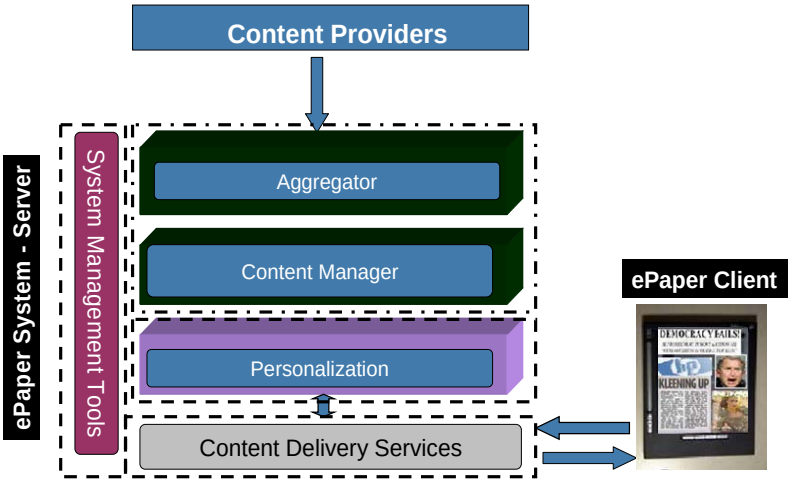

Figure 1. General architecture of the ePaper

The *Aggregator* interacts with content (news) providers and imports news item to the ePaper repository. (It may be assumed that the management of the ePaper service has business arrangements with certain providers. The business models with the providers, as well as with the clients/users, are beyond the scope of this project, and are immaterial for the description of the system.) A news item obtained from a provider may consist of one or more text and image files in certain formats, and include various metadata. For example, Reuters uses the NewsML format and its specialized metadata structures. (NewsML - News Markup Language - is a standard for the exchange of news supported by the IPTC (www.iptc.org). The main responsibility of the aggregator is to check the content providers for new news items, download them, create an index for each, and store the new aggregated items in the ePaper's file system.

The *Content Manager* processes the content of each news item received from the aggregator, and prepares it for personalization and delivery to relevant users. Its main responsibility is classification of the items: A text classification algorithm is used to analyze the content of each item and determine the concepts that best represent it. For this, the system maintains a hierarchical news ontology, which is based on the IPTC Subject Codes taxonomy. (More details on the Content Manager and the classification process in subsequent sections.)

The *Personalization* layer consists of a novel personalization engine that determines the level of relevancy of each news item to each user. The personalization engine applies content-based and collaborative filtering algorithms. The content-based filtering algorithm computes the similarity of the ontology concepts that represent each item, to each user's profile. A user's profile too consists of ontology concepts, which are initially defined by the user (upon registration to the system), and later on are dynamically updated by the system based on implicit user feedback. The measure of similarity between an item's and a user's profile considers the ontological proximity (or distance) between concepts in the two profiles. The collaborative filtering algorithm determines the similar users of each user (based on how many common items they have read), and computes a time-factor which considers how long ago (in hours) each item was read by each user. The final relevancy score of an item is computed as a weighted average of the two filters' rankings, taking into consideration the "maturity" of each item in the system: the more readers an item had, the more weight is given to its collaborative score. The result of the personalization process is a ranked list of items that will be delivered to each user, classified within the main ontology concepts that are of interest to her. The user can ask the system to refresh the items list anytime; as result, more news items may be delivered, and the ranking of all items on the reading device may be updated according to the fresh personalization process. The user can overrule the personalization engine, either by asking to get a "standard edition" of a certain newspaper, or by browsing the news items that exist in the repository, using menus of ontology concepts.

The *Content Delivery Services* layer orchestrates the processes of the system. It interacts with the *Personalization* layer, submits requests for personalized news from users, and sends the ranked news items it receives to the user. It also receives feedback from the user (tracking user's behavior data) and sends this data to the *Personalization* layer, which updates the user's profile to reflect the recent user's reading preferences.

The *System Management Tools* layer provides standard system tools such as logging and reporting, as well as special tools for the ePaper application. This includes, among else: a) Ontology Editor: this tool enables maintenance of the ontology, i.e. adding new concepts to the ontology, as new concepts may evolve over time. b) Registration subsystem: this is a web-based system where each new user registers and subscribes to preferred ePaper services, including: 1) some demographic and billing information (which are of no interest here); 2) selection of favorite content providers (i.e. newspaper) from whom to receive news; 3) an option to select a "standard" edition of a newspaper (instead of a personalized one); 4) an option not to track the user's reading, which my be desirable by certain users, and by that avoiding implicitly update of the user's profile; 5) definition of an initial content-based profile, by selecting concepts from the hierarchical ontology and determination of their initial weights of importance. As said, the initial profile will be updated dynamically according to implicit feedback from the user's reading device.

The Client system interacts with the *Content Delivery Services* layer for receiving data. The data sent to a user includes the user's profile information and a ranked-list of news items as determined by the *Personalization* layer. The Client is in charge of rendering the content and adapting it to preferred layout, and presenting the content to the user. To manage the variety and constraints of different mobile devices, the system supports dynamic content adaptation mechanisms based on the user's device, the user's preferences and local customizations made by each user. Thus, the presentation of content functionality is loosely coupled with the content preparation process, a capability that may scale the number and variety of devices supporting this service easily.

3. Content Management

Figure 2 presents the *Content Management* layer. It receives news items from the *Aggregator*, parses each item to extract content and metadata, classifies its content, and stores it in a repository of "active" items, to be delivered to users. (Items that are not read for some while by users are archived and become inaccessible from the mobile device). The *Content Management* layer consists of several units; some of them are described below.

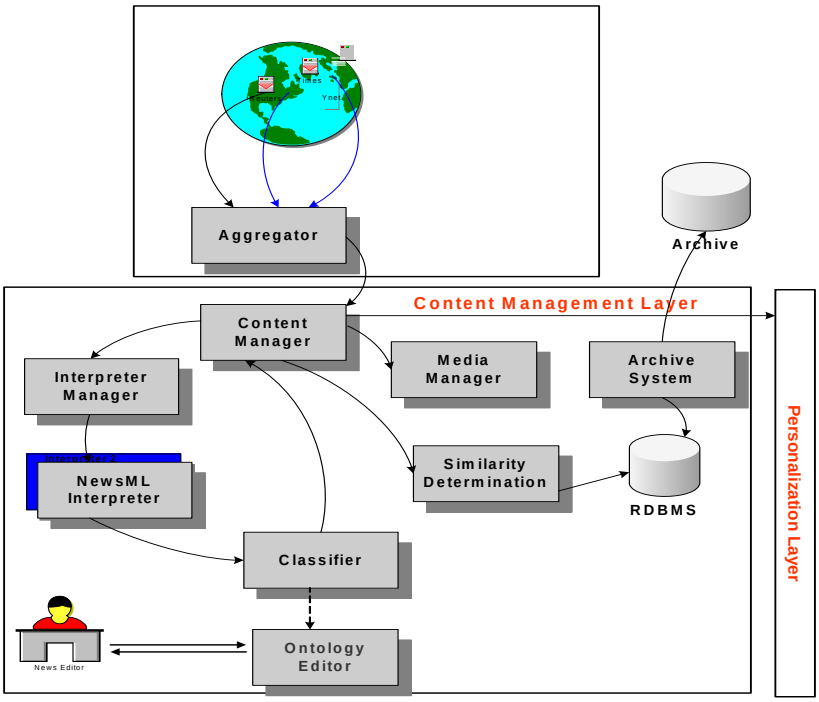

Figure 2. Content Management layer

3.1. The Content Manager

The Content Manager is orchestrating the processes of the *Content Management* layer. It receives news items from the *Aggregator* and sends them to the Interpreter Manager. Then it receives classified data back from the Classifier and sends them to the other functional units of the *Content Management* layer. After an item passes all the functional units, it is stored in the repository of "active" items, ready to be used by the *Personalization* layer.

3.2. The Interpreter Manager

The ePaper system is able to handle news items coming from multiple news providers, in multiple languages and in multiple formats. The Interpreter Manager is responsible for identifying the item's format and activating an appropriate interpreter, i.e., the interpreter that is able to "understand" the item's format and language, and extract from it the relevant metadata. The metadata it extracts from the item includes e.g., the item's provider/source, language and date of creation. Then it passes the item along with these metadata to the Classifier.

Currently, two interpreters have been implemented in the ePaper prototype system: one for NewsML format, which is used by many news providers, e.g. Reuters. The other interpreter is for RSS format. The ePaper can easily be extended to handle other standard formats by developing dedicated interpreters to each standard.

3.3. The Classifier

The ePaper system uses a news-ontology as a common language for content-based filtering. The ontology concepts are used to represent the news items' profiles and the users' profiles; the content-based filter will measure the similarity between the two profiles to determine the level of relevancy of each item to each user.

The ontology of ePaper is based on IPTC NewsCodes. NewsCodes is a set of controlled vocabularies; among other, it includes a Subject ontology, which consists of about 1400 concepts, organized in a 3-level hierarchy (termed Subject, SubjectMatter and SubjectDetail). News-providers who use NewsML use these concepts to describe the content of their articles, including them as part of the metadata attached to each news items.

For example, some of the first (Subject) level concepts of the IPTC Subject codes taxonomy are Sport, Politics, Economy, Education, Health and Science and Technology. Some of the second (SubjectMatter) level concepts of Politics are Election, Diplomacy, Defense, Government, and Parties. Some of the third level (SubjectDetail) concepts of Diplomacy are Summit, International Relations and Peace Negotiations.

The Classifier is responsible for determining the ontology concepts that will represent each news item and their weights, i.e., to define its content-based profile. A news item may deal about more than one concept; hence, we are dealing with a multi-label classification problem, where a news item may be classified into many concepts and in different levels of the ontology hierarchy. For example, a news item about an attempt of assassination at a presidential elections campaign can be classified to a number of concepts, such as Elections (which is a sub-concept of Politics) and Crime.

The classification process will be described in more detail in Section 5; before that, in the next section we provide a brief overview and related work on classification

4. Related Work on Classification

First, we provide a general introduction on text classification; then we present related work on news classification.

4.1 Text Classification

Text Classification or Categorization (TC) is the task of automatically assigning a text document (in our case, a news) to one or more predefined categories (in our case, ontology concepts) based on its contents. Nowadays, the dominant approach in TC is Machine Learning [Sebastiani, 2002]. According to this approach, a general inductive process automatically builds a text classifier by learning, i.e., by observing the characteristics of a set of previously classified documents - a training set. These characteristics are then used to classify new documents.

Different types of TC tasks can be distinguished: From a category assignment point of view, we distinguish between single-label and multi-label classification. In Single-label (also called multi-class) TC, exactly one category must be assigned to a document. In Multi-label TC, any number of categories may be assigned to a document. Binary categorization is a special case of single-label categorization, in which there is only one category and each document can be assigned to it or not ("yes", "no").

TC tasks can also be differentiated by the structure of the predefined categories set. In Flat categorization, the predefined categories are treated in isolation and there is no structure defining the relationships among them. Most of the studies in TC have focused on flat classification, and after many years of research flat classification has become a well-established research area and many good classifiers have been developed [Sun and Lim, 2001]. In Hierarchical categorization, the predefined categories are organized in a hierarchical structure that reflects relations between them. Most hierarchies are organized in tree-like structures, i.e., there are parent-child relationships between categories. In hierarchical classification, we can distinguish between cases where all documents belonging to a child category also belong to the parent - called *strong subsumption*; and cases where a child category has documents that do not belong to its parent category - called *weak subsumption*.

4.2 Multi-Label Classification

Many classification methods, such as Naïve Bayes, SVM, and Logistic Regression, are of the single-label type. Research on multi-label classification has received much less attention. Some methods that are mainly used for multi-label classification are presented below [based on Tsoumakas and Katakis, 2007].

The most popular approach for multi-label classification is binary approach (also called one-against-the-rest). A separate classifier is learned for each category C_i. The original data set is transformed into |C| data sets. The data set for each category C_i contains all examples of the original data set, labeled as c if the labels of the original example contained c, and as ¬c otherwise. For the classification of a new instance x, this method outputs as a set of labels the union of the labels predicted by the |C| classifiers.

This method has two main problems. First, it assumes independence of categories, which is not always true; there may be strong dependence between categories, in particular in hierarchical classification. Also, relations between categories on the same level can exist. For example, the following categories have some dependency: 'Politics' and 'Unrest, Conflicts and War'; 'Environmental Issue' and 'Health'. In such cases, association of an item with one category may influence its probability to be associated with a related category. But the binary approach cannot model such relations. The second problem is that a big number of binary classifiers have to be learned, which may cause memory problems, and take a lot of time because each new instance should be processed by all |C| classifiers.

Another, less popular approach, in multi-label classification, is to consider each different set of labels that exist in the multi-label data set as a single label. It so learns one single-label classifier for C` categories, where C` is

the power set of initial C categories. One of the negative aspects of this method is that it may lead to data sets with a large number of classes and few examples per class.

Another, yet not well-documented method is to learn one multi-class classifier, which can output a distribution of certainty degrees (or probabilities) for all labels in C, and then post-processes this distribution to output a set of labels. One simple way is to output labels for which the certainty degree is greater than a specific threshold (e.g. 0.5). A more complex way is to output labels for which the certainty degree is greater than a percentage (e.g. 70%) of the highest certainty degree. This method too ignores possible dependencies among categories.

4.3 Hierarchical Multi-Label Classification

Hierarchical classification has two main advantages compared to flat classification. First, it enables easy location of required categories when there are a significantly large number of categories; it is much easily to search among some high-level categories and then among some related sub-level categories, than to perform a general search among all existing categories. Second, it reflects the intuition of relatedness of topics that are close to each other in the hierarchy.

Two approaches were adopted by existing hierarchical classification methods: big-bang, and top-down level-based approach. In the big-bang approach, a document is classified (or rejected) into (or from) a category in the category tree by a classifier in one single step. In the top-down level-based approach, one or more classifiers are constructed at each level of the category tree, and each classifier works as a flat classifier at that level. A document will first be classified by the classifier at the root level into one or more lower level categories. It will then be further classified by the classifier/s at the lower level category/ies, until it reaches one or more final categories, which can be leaf categories or internal categories. (A classifier can stop at an internal node if an item cannot be classified to any of its child categories.)

One of the important works on hierarchical TC is [Koller and Sahami, 1997], who divide the hierarchical classification task into a set of smaller classification tasks, each of which corresponds to some split in the classification hierarchy. They show that this approach enables obtaining significantly higher accuracy compared to a single massive classifier.

4.4 Classification in the News Domain

In this section, some implementations of classification algorithms in news domain are presented.

An interesting implementation of multi-label classification is presented in [Antonellis et al, 2005]. They implemented a news categorization system that decomposes each document into its sentences, computes term to sentences matrix and performs multi-label classification by estimating the similarity of each sentence to the category vectors. Category vectors are computed by combining the columns of the corresponding term to sentences matrix of the training set. If the estimated similarity is above a threshold defined during the training phase, the document is classified to the corresponding category. Thus, multi-label classification is allowed.

[Calvo et al., 2004] applied automatic Naïve Bayes classifier on news stories from Reuters RCV1 Corpus. They performed flat multi-label classification using two different thresholding strategies: score-based, where all categories with a score above some threshold are assigned to a document; and rank-based, where the k top-ranked categories are assigned to a document. The results were compared to a kNN classifier. It was shown that for the rank-based thresholding strategy, the best performance was achieved for k = 1. The explanation is that in this data set, the average of categories that each document is assigned to is approximately 1. It was also noted that the best performance of the rank-based strategy is considerably worse than the score-based thresholding strategy. Regarding the comparison to kNN classifier, it was concluded that Naive Bayes classifiers produce lower quality classification but seem to be better suited for applications were classification needs to be performed at real time; kNN instead produces better classification but places to much load on classification time.

One of the examples for implementation of hierarchical classification in the news domain is presented in [Eilert et al., 2001]. The developed system called Bikini was supposed to classify news from several WWW news sources into concepts hierarchy (ontology). Bikini's ontology was handcrafted and included approximately 130 categories, with a maximum three-depth of four. For classification purpose, it was interpreted as a flat list of concepts. Any node in the tree was interpreted as a leaf tree. For intermediate nodes this was achieved by introducing an artificial ‘miscellaneous’ successor node. The classification was performed by comparing document representation vector with categories representation vectors using simple similarity measure.

5. The ePaper Classifier of News

In ePaper, we implemented hierarchical multi-label classification algorithm utilizing a flat multi-class classifier provided by LingPipe open source software [LingPipe]. The utilized LingPipe's LanguageModel (LM) Classifier is based on statistical language modeling techniques and performs probability-based classification into non-overlapping categories.

The motivation for language modeling has traditionally come from speech recognition; recently it became widely used in many other application areas. Language modeling aims at predicting the probability of naturally occurring word sequences, s = w1w2…wn. It puts high probability on word sequences that actually occur, and low probability on word sequences that never occur. The simplest and most successful approach to language modeling is based on the n-gram model [Peng, 2003]. An n-gram is a sub-sequence of n items from a given sequence. The items can be characters or words. If the language model is based on character sequences, it is called *character* language-model. According to language modeling approach, the probability of any word or character sequence is calculated as frequency of the observed patterns

$$\Pr(w_i|w_{i-n+1}...w_{i-1}) = \frac{\#(w_{i-n+1}...w_i)}{\#(w_{i-n+1}...w_{i-1})}$$

where #( ) denotes the number of occurrences of a specified gram in the training corpus.

The LingPipe's LM classifier constructs a character language-model for each category during the training phase; then, at classification time, it calculates conditional and joint probabilities of each category for the classified object. Also, a score, which is the character cross-entropy rate normalization allowing between-document comparisons, is provided. This score is ordered in the same way as the joint probabilities. Finally, LingPipe classifier returns one best category as result of classification process.

It was found in many researches that the language modeling approach provides competitive and often superior results compared to more sophisticated learning techniques [Peng, 2003]. Moreover, it provides scalable training and classification time performance. Therefore, we utilized LingPipes LM Classifier, allowing classification into one of non-overlapping categories, and enhanced it to satisfy our hierarchical and multi-label requirements.

For multi-label classification, we apply an approach based on estimations of a posteriori probabilities of an item to belong to some category. According to this approach, we can think of a classification as "better estimated" if the probability of the destination category is above some threshold. For example, if the probability of an item D to belong to category C is 0.9 it is better estimated than if it is only 0.4. To determine the threshold for multi-label classification we use the cross-entropy scores provided by LingPipe classifier, as they are better suited for cross-document comparison. Using this principle, we classify item D in n categories Ci, such that

$$score(Ci,D) > \tau \; ; \; 1 \leq i \leq n \text{ and } 1 \leq n \leq |C|$$

where the threshold τ is learned on empirical runs. Then, the set of categories assigned to item D is {C1,…,Cn} a set of "n best" categories with classification score above the defined threshold.

We chose this approach as it has shown favorable results compared to the more standard binary approach to multi-label classification [Vilar et al, 2004]. Moreover, it is very scalable in terms of memory and time performance: it does not require additional classification models to be loaded into the memory and an additional classification to be performed at run time. Also, it natively ranks the categories assigned to an item according to their level of relevance; this rank is later used by the personalization and content delivery services of ePaper.

For hierarchical classification, we apply the top-down level-based approach. We implemented this approach by constructing a separate classification models for each non-leave concepts of the ontology tree. At the first level, there is one model for classification into one or more of 17 Subject concepts. At the next level, there are 17 models for classification into the SubjectMatter concepts; and about 130 models for classification at the third (SubjectDetail) level of the ontology. Hence, the number of models generated is identical to the number of non-leave concepts in the hierarchy plus one for the root node.

After all the above multi-label and hierarchical extensions, the classification of a new item is performed as follows: First, the item is classified into one or more top level (Subject) concepts. Then, if one or more of the Subjects are assigned to the item, it is further classified into one or more child concepts (SubjectMatter) of the Subjects. Then, if one or more of SubjectMatter concepts are assigned to the item, it is further classified into their child concepts (SubjectDetail). The classification process stops when classification to the detailed concept is not confident enough. (Initial confidence thresholds are defined by configuration parameters, depending on empirical runs.)

Once the results of the classifications at each level are obtained, the final classification is determined according to the received concepts' weights and defined confidence thresholds. The most specific concepts having classification scores above the pre-defined threshold are assigned to the item.

The following example will clarify the classification process. Suppose a certain news item has to be classified. First, it is assigned by the root classifier to one or more of the top-level concepts; assume it is assigned to Economy and Science & Technology. Then, the second level classification starts for each of these concepts. The Economy and Science & Technology classifiers are invoked one after the other, attempting to classify the item to one or more of their sub-concepts. Assume that Macro Economics and Energy & Resources concepts are returned by the Economy classifier, and Applied Science concept is returned by the Science & Technology classifier. When Macro Economics and Energy & Resources concepts are received, the third level classification starts for each of them. The Macro Economics and Energy & Resources classifiers are applied on the item one after the other, attempting to classify it to one or more of their sub-categories. The same is for the Applied Science concept: its classifier is applied on the item attempting to classify it to one or more of its sub-categories. Assume that the results of the third level classification are as follows: Macro Economics classifier returns Government Aid concept; but Energy & Resources and Applied Science classifiers cannot classify the item to any of their sub-categories (because the results are below the defined thresholds).

The weight of each assigned concept is calculated based on its cross-entropy score received at the corresponding classification level. Assume the scores of the assigned concepts are as follows: Economy (-1.65), Science & Technology (-1.78), Macro Economics (-1.73), Energy & Resource (-1.75), Applied Science (-1.84), Government Aid (-1.80); and the defined threshold is (-1.86). The item is classified to the most specific concepts according to their location in the hierarchy, i.e., Government Aid, Energy & Resources and Applied Science. We then convert each concept's score onto a weights scale of 0--1, where a concept's weight expresses its absolute importance in the item. This is done as follows: the "best" concept gets the weight 1; the weights of the other concepts are lower, proportionally. In our example, the following concept–weight pairs are assigned to the item: Energy & Resource, 1 (= -1.75), Government Aid, 0.97 (= -1.75/-1.80), and Applied Science, 0.95 (= -1.75/-1.84).

6. Summary and Future Issues

The ePaper prototype system is now undergoing various evaluations. This includes evaluations of the content personalization algorithms and the content adaptation to the reading device and users' preferences. Regarding content personalization, we examine the effect of various parameters of the content-based and the collaborative filtering algorithms, e.g. the optimal scores of the various types of ontological similarity between user and item profiles; the optimal number of concepts to be considered in a user's profile; and the optimal method to update a user's profile based on implicit feedback. For this, we run controlled experiments with users who evaluate the relevancy of news items provided to them, and compare their evaluations to the system's ranking of those items,.

Regarding the classification process, additional work is required for tuning threshold parameters. For this, a combination of multi-label and hierarchical measures will be defined, that will take into consideration specific system requirements. We plan to compare the results of the adopted 'thresholding' strategy to the common binary approach to multi-label classification. Since the construction and classification using about 1400 binary classifiers seems unacceptable for the ePaper system (because of time required for each classification), perhaps the binary approach will be applied only on the top level of the hierarchy (i.e. for the 17 Subject concepts). A method for multi-label classification taking into in consideration dependencies among concepts is also under development.

Acknowledgments

The ePaper project was sponsored by Deutsche Telekom Co., and developed at the Deutsche Telekom Laboratories at Ben-Gurion University.

Bibliography

[Antonellis et al., 2005] Antonellis, I., Bouras, C. and Poulopoulos, V. (2005). Personalized news categorization through scalable text classification. 8th Asia Pacific Web Conference (APWEB '06).

[Eilert et al., 2001] Eilert, S., Mentrup, A., Мьller, M.E., Rolf, R., Rollinger, C.R., Sievertsen, F. and Trenkamp, F. (2001). Bikini: user adaptive news classification in the World Wide Web. Workshop on Machine Learning for User Modeling; 8th Intl. Conf. on User Modeling.

[Calvo et al., 2004] Calvo, R.A., Lee, J.M. and Li, X. (2004). Managing content with automatic document classification. J. Digit. Inf. 5 (2).

[Koller and Sahami, 1997] Koller, D. and Sahami, M. (1997). Hierarchically classifying documents using very few words. Proceedings of the 14th International Conference on Machine Learning, pp.170-178.

[LingPipe] LingPipe – a suite of Java libraries for the linguistic analysis of human language. http://www.alias-i.com/lingpipe/index.html

[Eilert et al., 2001] Peng, F., Schuurmans, D. and Wang. S. (2003). Language and task independent text categorization with simple language models. Proceedings of HLT-NAACL 2003.

[Sebastiani, 2002] Sebastiani F. (2002). Machine learning in automated text categorization. ACM Computing Surveys, Vol. 34 (1), pp.1-47.

[Sun and Lim, 2001] Sun A. and Lim E.P. (2001). Hierarchical text classification and evaluation. First IEEE International Conference on Data Mining, ICDM'01, pp. 521-528.

[Tsoumakas and Katakis, 2007] Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: an overview. International Journal of Data Warehousing and Mining, Vol. 3 (3), pp. 1-13.

[Vilar et al., 2004] Vilar, D., Castro, M.J. and Sanchis, E. (2004). Multi-label text classification using multinomial models. Proceedings of 4th International Conference on Advances in Natural Language Processing (EsTAL 2004), pp. 220-230.

Authors' Information

Lena Tenenboim (Dept. of Information Systems Engineering - ISE, Ben-Gurion University, Beer-Sheva 84105, Israel; e-mail: lenat@bgu.ac.il) is Ph.D. student. She holds a M.Sc. in ISE from Ben-Gurion University. Her research interests include Information Retrieval and Filtering, Machine Learning and Data Mining.

Bracha Shapira (Dept. of ISE, Ben-Gurion University, Beer-Sheva 84105, Israel; e-mail: bshapira@bgu.ac.il) is Senior Lecturer of ISE. She holds a M.Sc. in Computer Science from the Hebrew University in Jerusalem and a Ph.D. in Information Systems from Ben-Gurion University. Her research interests include Information Retrieval and Filtering, specializing in various aspects of user profiling and personalization. In addition, she has worked on privacy preservation while browsing and on formal models of Information Retrieval systems. She is leading research projects in these domains at the Deutsche-Telekom Research Lab at Ben-Gurion University.

Peretz Shoval (Dept. of ISE, Ben-Gurion University, Beer-Sheva 84105, Israel; e-mail: shoval@bgu.ac.il) is a Professor of ISE. He earned his Ph.D. in Information Systems from the University of Pittsburgh, where he specialized in expert systems for information retrieval. In 1984, he joined Ben-Gurion University, where he founded the Information Systems Program, and later founded and headed the Dept. of ISE. His research interests include information systems analysis and design methods, data modeling, and information retrieval and filtering.