INDONESIAN NEWS HARVESTER AND RECOMMENDER SYSTEM

Adi Wibowo[1], Rolly Intan[2], and Nydia Valentina[3]

[1,2,3]Informatics Department, Faculty of Industrial Technology, Petra Christian University
Jln. Siwalankerto 121-131, Jawa Timur 60236, Indonesia
[1]adiw@petra.ac.id; [2]rintan@petra.ac.id; [3]nydia@luke.petra.ac.id

Received: 19th August 2017/ **Revised:** 20th September 2017/ **Accepted:** 25th September 2017

Abstract **-** To provide convenience for the user that frequently read the news, a system to gather, classify, and provide news from several news websites in one place was needed. This system utilized a recommender system to provide only relevant news to the user. This research proposed a system architecture that used vector space model, and Rocchio relevance feedback to provide specific news recommendation to user's feedback. The results are that the proposed system architecture can achieve the goal by using five levels of feedback from the user. However, the time needed to gather news is increasing exponentially in line with the number of terms gathered from articles.

Keywords: news harvester, recommender system, vector space model, Rocchio relevance feedback

I. INTRODUCTION

There are applications that gather news from several digital news outlets and provide them to users such as Pulse, Flipboard, Feedly, and Digg Reader. The advantage of this application is that users do not need to visit each news website. The application will harvest news articles from each news websites, classify them, and provide an easy tool for users to read the news that they need. Usually, these applications gather news from international news outlets such as BBC, CNN, and others. To increase user engagement, each application utilizes a recommender system.

A recommender system is defined as a system that collects information about items. This information creates user's preferences. From these preferences, a recommender system can provide predictions or recommendations of other items (Bobadilla, Ortega, Hernando, & Gutiérrez, 2013). This system provides recommendations through four types of filtering. Those are content-based filtering, collaborative filtering, demographic filtering, and hybrid filtering.

Content-based filtering uses the description of items' representation as dataset to be compared with user's preferences (Pazzani & Billsus, 2007). User's preferences can come from the ratings that the user gives for each item. Content-based filtering has a weakness that the dataset must have enough information in items' representation to distinguish the different items, and between items that user finds interesting and uninteresting. The other weakness of content-based filtering is that because this filtering system primarily uses feedbacks from users, when there are many items to be rated by users, the user may miss it.

To address content-based filtering weakness, a recommender system can use collaborative filtering. Collaborative filtering assumes that people with similar taste will give similar rating to items (Hameed, Jadaan, & Ramachandram, 2012). This system aggregates rating from several users, identifies similarity of users based on those ratings, and provides recommendations to a user using the similarity information. It also has an advantage compared to content-based filtering. It uses users' ratings, not items' content. Thus, it can be used to items that have simple or little information in its representation dataset. This situation also benefits this filtering system as it usually does not need to concern about language, stemming, or parsing. Meanwhile, the weakness of collaborative filtering is there are sparse data about users' preference. This situation will create clusters that cannot represent users' profile. The other weakness is when there are many users or items. The computational resources to filter those will be needed compared with content-based filtering.

Moreover, demographic filtering uses the principle that people who share similar attributes have common preferences and will rate items similarly. This filtering system uses demographic data in user's profile like gender, age, or country (Safoury & Salah, 2013). This demographic data can be used together with another filtering, like collaborative filtering, to form a hybrid filtering. The other method of hybrid filtering was proposed by Said, Plumbaum, De Luca, and Albayrak (2011). For users with the same demographic data, the level of similarity between users would be multiplied by a very large multiplier factor. This process of multiplication would not be performed on users who had no demographic similarities.

Next, hybrid filtering combines another filtering system to gain advantages from one of filtering methods and eliminate the weaknesses of others (Burke, 2002). There are several approaches to construct a hybrid filtering such as weighted, switching, mixed, feature combination, cascade, feature augmentation, and meta-level method. The weighted hybrid method combines scores from two or more filtering into one recommendation. One of the weighted methods is to give equal weight of score from content-based filtering, and collaborative filtering, and adjust the weight when the users give more ratings. Meanwhile, the switching method does not combine scores of filtering, but it only picks one filtering results according to some criteria. Then, mixed method combines techniques to present more than one filtering approach altogether. The feature combination method primarily uses content-based filtering, but it also includes information in dataset from collaborative filtering. The cascade method usually conducts one filtering approach first and refines the results by using another filtering approach. In meta-level hybrid method, a filtering approach produces models to be used by another filtering approach.

Several systems have aggregate news from Indonesian news sites and provide it to user in one place. The most popular applications are Baca, BaBe, and Kurio. They utilize content-based recommender system that uses implicit feedback in the form of history of what news the user has read. However, it is not from explicit feedbacks such as user's rating.

This research proposes an application system architecture that uses both implicit and explicit feedbacks, and content-based filtering to provide recommendations. Content-based filtering is used in this research because news articles are texts, and each article can consist of thousands of words. Thus, the weakness of content-based filtering does not apply in this case. Moreover, this architecture utilizes a Rocchio algorithm supported by Vector Space Model to analyze user's preference, gather news articles from several Indonesian news websites, and provide specific news articles according to the user's preference.

II. METHODS

This research uses Waterfall method to construct the prototype. The prototype is used to test if the proposed architecture that will be explained can be implemented. The first step is to gather information about recommender system and possible news syndication format (Really Simple Syndication (RSS), Atom, Document Object Model (DOM)). For the second step, this research conducts surveys on six Indonesian news websites to determine the news syndication format. Next, in the third step, the researchers analyze and design system architecture into four server components, and one Android application used by the users. The fourth step is to develop architecture blocks and Android applications according to the design. The last step of Waterfall method is to test four server components and Android application. The objective of the tests is to measure time needed to perform indexing step and check if the system architecture can gather news from several news outlets. It also runs android application to see whether feedbacks gathered from the user can change article recommendations that match with those feedbacks.

III. RESULTS AND DISCUSSIONS

The architecture of the proposed system is shown in Figure 1. This system includes four components such as article grabber, VSM indexer, recommender system, and article service provider. Article grabber collects news from several news websites and stores it in the database. Meanwhile, VSM indexer will count the weight of terms in each news article and use it to build vector models. Next, a recommender system uses preferences from the users and combine it with user recommendations for each news article to build a list of suitable articles for each user. Article service provider will send the list and articles' content to an Android application. Android application is used by the users to register into the system, choose categories of the news, read the list of news, read a particular news article, and submit feedback to the system.

Article grabber gathers news from several news websites, parse the content, and store them into database. In this research, the news articles from six Indonesian news websites are gathered. Those are Detik (http://www.detik.com), Kompas (http://www.kompas.com), Okezone (http://www.okezone.com), Kapanlagi (http://www.kapanlagi.com), Liputan6 (http://www.liputan6.com), and BBC Indonesia (http://www.bbc.com/indonesia).

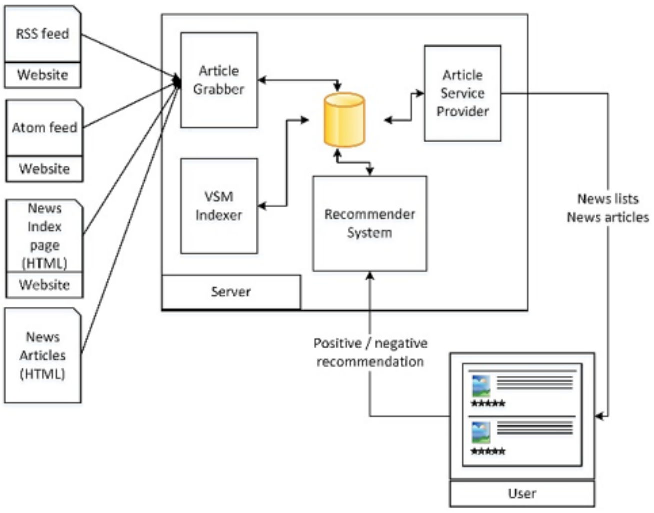


Figure 1 System Architecture

In the news websites, article grabber performs three steps. First, article grabber identifies what platform the news website uses to broadcast the lists of articles. The platform can be RSS, Atom, or using specific HTML like DOM structure. In the second step, article grabber downloads and parses RSS files, Atom files, or HTML in index page to obtain all links of articles. To parse RSS and Atom files, this system uses XML parser. Meanwhile, to parse HTML files, it uses HTML DOM Parser. In the third step, article grabber retrieves its HTML document and parses it to obtain title, summary, content, author, date of publications, image, and categories. This step is repeated for each link gathered from the second step. The method used to parse document content is HTML DOM Parser.

Detik, Okezone, and Kapanlagi use RSS 2.0 as the platform to share their feeds. BBC Indonesia uses Atom. However, Kompas and Liputan6 do not provide feeds so article grabber must parse the articles through DOM in index page to find the title and link of news articles. The weakness of using DOM to parse HTML page for Kompas and Liputan6 is if the administrator of these two websites changes HTML layout, DOM parser for this website also needs to be changed. Table 1 shows the elements gathered on the news website.

Table 1 Elements gathered on the News Websites

Elements	Detik	Oke zone	Kapan lagi	BBC	Liputan6	Kompas
Title	V	V	V	V	V	V
Summary	V	V	V	V	X	X
Content	V	V	V	V	V	V
Author	V	X	X	V	X	X
Date	V	V	V	V	V	V
Image	V	V	X	V multiple	V	X
Category	X	V	V	V multiple	X	X
Link	V	V	V	V	V	V

After news articles from all six Indonesian news websites are downloaded and parsed, VSM indexer prepares the terms and weight database in the terms. VSM indexer uses cosine similarity proposed by Gerard Salton (Turney & Pantel, 2010) to find similarity between news articles and user's preferences. User preferences in this research consist of a list of article categories that the users select when they register to the system. The users also list the interesting and uninteresting news articles.

To find similarity between user preferences and news articles, the system must construct a vector for each news article collected by article grabber. The vector is constructed from the weight of terms in news articles. To construct vectors to represents the article, this system needs to identify the terms from the news articles. The researchers perform three steps. The first step is parsing each term from news articles. This step does not try to identify terms in the form of phrases from articles but in only words. This approach has a weakness. The popular phrases like names of actors or actresses, names of established institutions such as Komisi Pemberantasan Korupsi (KPK), or other phrases will not be recognized. However, this approach is chosen to simplify and speed up the parsing process because the news website can produce hundreds or thousands articles every day.

The second step is article grabber identifies whether the words are in English or Bahasa Indonesia. To do this identification, a simple lookup into English and Bahasa Indonesia database is performed. If the words are in English, the system will check whether the word is one of English stop words or irregular verbs. If the word is a stop word, it will be deleted from system. However, if the English word is not an irregular verb, it will be included in stemming process using Porter stemmer (Karaa, 2013). An irregular verb is a verb that does not follow the rule of adding –d or –ed to create past tense. Because there are so many variations among irregular verbs, there are no rules that can be applied to stem these words. Meanwhile, Indonesian word will be stemmed using IDN Stemmer.

The third step, article grabber stores the words into database. In this step, removing stop words is necessary to reduce the number of terms that need to be processed by the system, and to increase recommendation quality. Stop word is a list of terms that have no meaning or appear in too many documents (Zaman, Matsakis, & Brown, 2011). The examples of stop word in English are 'a', 'the', 'an', 'for', or 'of'. Meanwhile, the examples of stop word in Bahasa Indonesia are 'di', 'ke', 'dari', 'bahwa', or 'pada'. To remove English stop words, this system uses stop word list from Gerard Salton and Chris Buckley, and for stop word in Bahasa Indonesia, it uses a list from Indonesian Grammar from Moeliono.

This step is also the stemming process to get the base form of the term or its root form. It usually is to reduce document index size (Karaa, 2013). The use of stemming is necessary so that the word 'bangunan', 'dibangun', 'pembangunan' can be interpreted as the same term. Thus, the recommender system will not be influenced by variations of the terms.

After the terms from news articles can be identified and stored into database, the weight of the terms can be calculated. Wij is a weight of term j in article 'di'. To count the weight, the common method used is the product of term frequency (tf) and inverse document frequency (idf) as shown in the equation (Ropero, Gómez, Carrasco, León, & Luque, 2012).

$$wij = tf\,i,j * IDFj \tag{1}$$

$$IDFj = \log\left[\frac{D}{df_j}\right] \tag{2}$$

$$wij = tfi,j * \log\left[\frac{D}{df_j}\right] \tag{3}$$

Term frequency tfi,$_j$ is the number of term j in the article 'di'. The higher the number of tfi,$_j$ is, the higher the weight (w$_{ij}$) of term j will also become. Inverse document frequency (idf) shows the relationship between the particular terms in the whole documents or articles. The higher the number of documents containing the term j is, the less unique term j becomes in all documents. This means that the weight of term j will become lower.

After weight of all terms has been calculated, the vector that represents each news article can be constructed. The vector from user's preferences is compared with the vector that represents the news articles by recommender system. The similarity between these two vectors is computed using similarity function from Vector Space Model (Turney & Pantel, 2010). Q is the vector from user's preferences, and Di is the vector from news article I.

$$Sim(Q, Di) = \frac{\sum_i w_{Q,j} w_{i,j}}{\sqrt{\sum_j w^2_{Q,j}}\ \sqrt{\sum_i w^2_{i,j}}} \tag{4}$$

The next component, a recommender system, aims to find articles that match the needs or preferences of a user. In a recommender system, the users will tell the system what items that the user is interested and uninterested in (Pazzani & Billsus, 2007). These preferences will generate user's profile. User's profile is used to recommend other documents that fit with user's profile.

The recommender system approach used in this system is Rocchio relevance feedback (Rocchio, 1971). This approach is illustrated in Figure 2.

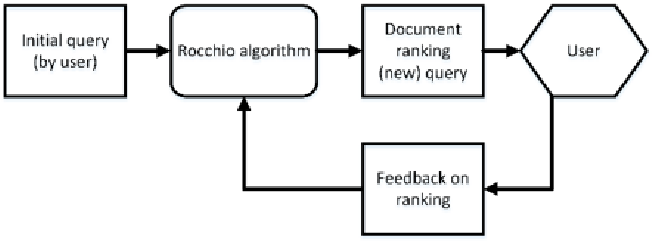


Figure 2 Rocchio Relevance Feedback

Figure 2 shows that a recommender system starts from an initial query. Initial query is a vector constructed from user's preferences. In this research, user's preferences are the category that the users select when they register for the first time. The weight of terms from category will construct initial query vector. Then, initial query vector is compared with all news articles vector collected by article grabber to find similarities between initial query and news articles. Thus, similar articles are presented to the user. The users provide feedback consisting of which news articles that the users like and dislike. Feedbacks from users are used to modify the initial query and to generate new recommendations. Rocchio algorithm is shown as follows.

$$\vec{q}_m = \alpha\vec{q}_0 + \beta\frac{1}{|D_r|}\sum_{\vec{d}_j\in D_r}\vec{d}_j - \gamma\frac{1}{|D_{nr}|}\sum_{\vec{d}_j\in D_{nr}}\vec{d}_j \tag{5}$$

q_0 is an initial vector created from categories chosen by the user. D_r is a set of articles considered as relevant by the user. On the other hand, D_{nr} is a set of articles considered as irrelevant. α, ß, and γ are weights for each component.

In this research, a user can give rating for the news article. The article rating values are +2 for really interesting article, +1 interesting article, 0,5 for normal article, -1 for uninteresting article, and -2 for really uninteresting article. The user can give different rating value for different article. Thus, Equation 5 has to be slightly modified to accommodate a situation where ß and γ will be different for the news article, D_j, ß, and γ will be multiplied first with each D_j before those are summed. Equation 5 becomes as follows.

$$\overrightarrow{Q_m} = \left(\alpha\cdot\overrightarrow{Q_0}\right) + \left(\frac{1}{|D_r|}\cdot\sum_{\overrightarrow{D_j}\in D_r}\beta\cdot\overrightarrow{D_j}\right) - \left(\frac{1}{|D_{nr}|}\cdot\Sigma_{\overrightarrow{D_k}\in D_{nr}}\,\gamma\cdot\overrightarrow{D_k}\right) \tag{6}$$

There are two ways for users to give feedbacks to recommender system, implicit and explicit feedback (Palanivel & Sivakumar, 2010). Feedback is implicit when the users are not aware that they give feedback input to the system about the item. In addition, feedback is explicit when the users consciously provide data about which items interest the users and which does not. Explicit feedback usually gives inconveniency to the user because the user needs to do some work. However, it is a reliable indicator of what user thinks about the item. To modify initial query to create complete profile of the user, this research uses both implicit and explicit feedbacks.

To gather implicit feedbacks, the system will monitor which news categories that the user read the most and assume these categories as important for the user. Moreover, terms from categories are used to construct new initial query Q. Then, to gather explicit feedback, the user has the option to choose between 1 (dislike very much) to 5 (like very much) for the article that the user reads. User's choice will be converted into rating value number (+2, +1, 0,5, -1, and -2). Rating value number is set as ß and γ respectively for D_r and D_{nr}. To construct D_r and D_{nr} vector, this system will use term discrimination. Term discrimination is used because the relevant document (D_r) and irrelevant document (D_{nr}) consist of thousand terms. It will slow down the system. Term discrimination will find twenty terms from each document with the highest weight. These twenty terms will be used to construct D_r and D_{nr} vector.

Then, article service provider utilizes 13 AJAX functions as a data transfer interface between a server and Android application. AJAX is a platform to enable client/server communication (Singh, 2012). AJAX achieves this by using Javascript to send asynchronous data and uses DOM to trigger the transmission. 13 functions provided by article service provider are GetCategory (idmember: integer), ChangePassword (oldpass, newpass: varchar; idmember: integer), EditCategory (idmember: integer, edit_cat: array of integer), ViewNews (idnews: integer), LoadNews (idmember: integer), Login (user, pass: varchar), RateNews (idnews: integer), Recommend (idmember: integer), Register (pass, user, email, name: varchar; gender: char; birthday: varchar), SaveCategory (idmember: integer, chosen_cat: array of integer), SubCategory (), ViewByCategory (idcategory: integer), and ViewBySite(idsite: integer). Communication between Android application and the article service provider does not use Restful or SOAP methods, but it uses AJAX. AJAX is chosen because Android application is built using PhoneGap technology. It is easier to use AJAX in using PhoneGap manipulation of display element.

The server has functions for grabbing news articles from six Indonesian news websites, parsing and indexing news articles, and providing recommendations for users. Android application has functions for displaying categories, article lists and article content, and getting feedbacks from users.

From several experiments, Table 2 and Figure 3 show the time needed for each step of indexing from grabbing articles until VSM indexing. It shows that the time needed to process the increasing number of articles is not linear, but it is exponentially especially in the step of generating terms from articles.

Table 2 Time Needed for Each Step (Seconds)

No.	Steps	Number of articles				
		100	250	500	750	1000
1.	Generate Terms from Articles	4.117	8.277	15.826	30.438	55.964
2.	Count IDF	122	212	252	789	963
3.	Count Term Weight	396	1.183	1.888	4.841	6.461
4.	Term Discrimination	43	105	154	543	759

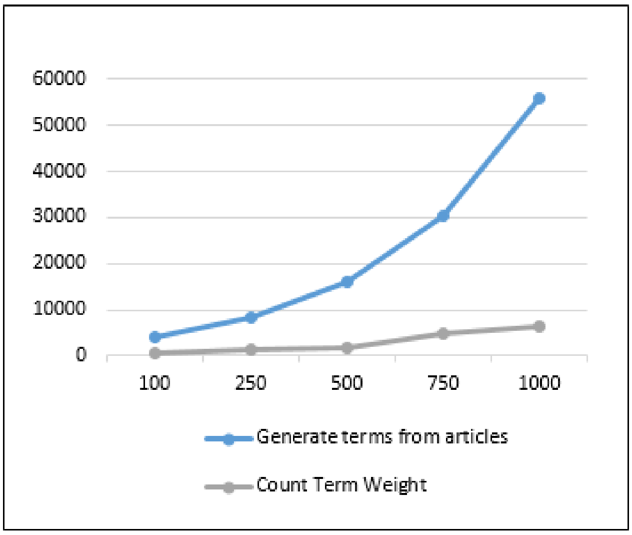
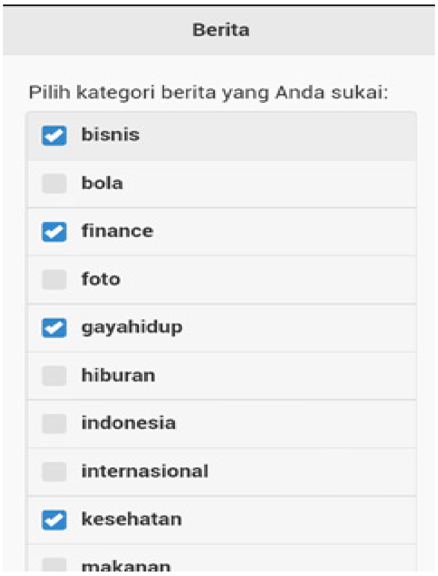


Figure 3 Time Needed for Each Process

To test if the proposed architecture works, the users register in Android application and provide feedback. If the system works as designed, the list of recommended articles will change according to user's feedback. There are four steps of testing. There are user's registration process, initial list of articles, feedback process, and modified list of articles according to user's feedback.

In the first step of testing, the users must register into the system. The users must choose which categories that interest the users as shown in Figure 4. These categories will become initial query vector. In the second step of testing, Android application can show several articles that match those categories that interest the users as shown in Figure 5. This list of articles is called initial list.

The third step of testing will be started if the users pick one of the articles from the initial list in Figure 5. The application can show the article's content and show feedback which is below the content. This process is shown in Figure 6. If the users pick "Dislike it so much" or "Dislike it", the recommender system will assume that the article is an irrelevant article. Otherwise, if the users pick "Like it", or "Like it so much", the recommender system will assume that the article is a relevant article for the users. The recommender system identifies twenty terms that represent the article and change Rocchio vectors that represent relevant or irrelevant articles. Then, the last step of testing shows that the recommender system can compute similarity of articles with a vector from Rocchio relevance algorithm. This step changes the article recommendation list according to user's feedback as shown in Figure 7.

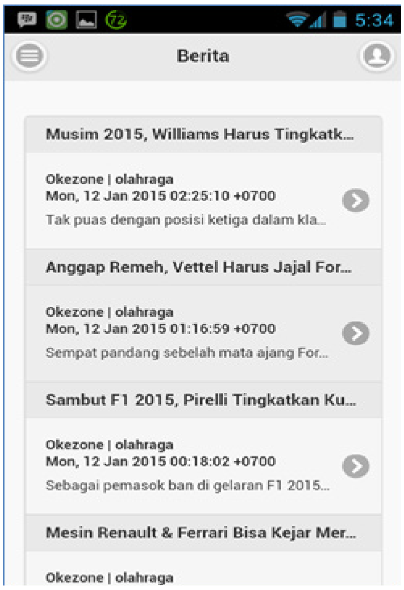


Berita

Pilih kategori berita yang Anda sukai:

- ☑ bisnis
- □ bola
- ☑ finance
- □ foto
- ☑ gayahidup
- □ hiburan
- □ indonesia
- □ internasional
- ☑ kesehatan
- □ makanan

Figure 4 Registration Form to Choose Categories to Capture User Interests



Berita

Musim 2015, Williams Harus Tingkatk...

Okezone | olahraga
Mon, 12 Jan 2015 02:25:10 +0700
Tak puas dengan posisi ketiga dalam kla...

Anggap Remeh, Vettel Harus Jajal For...

Okezone | olahraga
Mon, 12 Jan 2015 01:16:59 +0700
Sempat pandang sebelah mata ajang For...

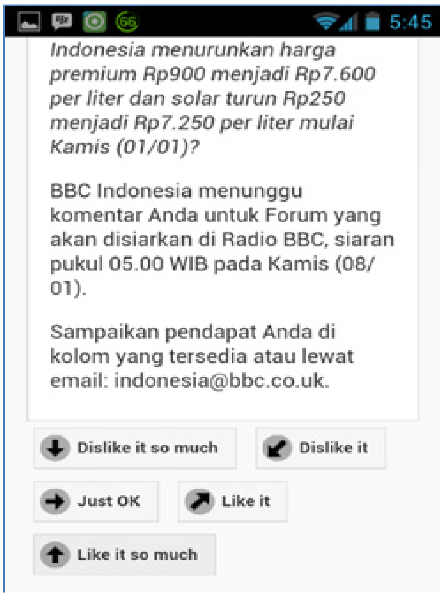
Sambut F1 2015, Pirelli Tingkatkan Ku...

Okezone | olahraga
Mon, 12 Jan 2015 00:18:02 +0700
Sebagai pemasok ban di gelaran F1 2015...

Mesin Renault & Ferrari Bisa Kejar Mer...

Okezone | olahraga

Figure 5 List of Articles that Match with User's Interests

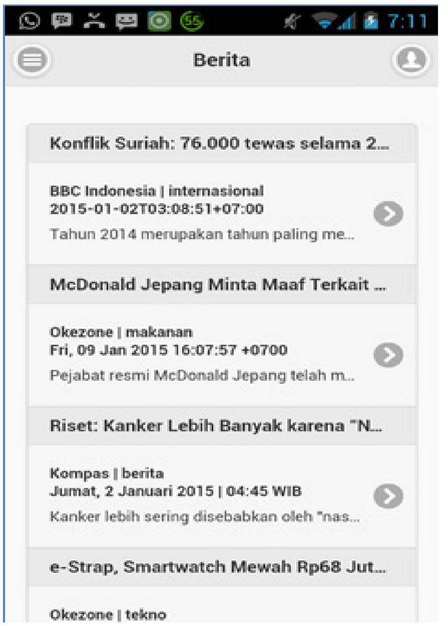


Indonesia menurunkan harga premium Rp900 menjadi Rp7.600 per liter dan solar turun Rp250 menjadi Rp7.250 per liter mulai Kamis (01/01)?

BBC Indonesia menunggu komentar Anda untuk Forum yang akan disiarkan di Radio BBC, siaran pukul 05.00 WIB pada Kamis (08/01).

Sampaikan pendapat Anda di kolom yang tersedia atau lewat email: indonesia@bbc.co.uk.

Figure 6 Article Content and Feedback Form



Berita

Konflik Suriah: 76.000 tewas selama 2...

BBC Indonesia | internasional
2015-01-02T03:08:51+07:00
Tahun 2014 merupakan tahun paling me...

McDonald Jepang Minta Maaf Terkait ...

Okezone | makanan
Fri, 09 Jan 2015 16:07:57 +0700
Pejabat resmi McDonald Jepang telah m...

Riset: Kanker Lebih Banyak karena "N...

Kompas | berita
Jumat, 2 Januari 2015 | 04:45 WIB
Kanker lebih sering disebabkan oleh "nas...

e-Strap, Smartwatch Mewah Rp68 Jut...

Okezone | tekno

Figure 7 List of Recommended Articles from the User's Feedback

IV. CONCLUSIONS

The proposed system architecture has succeeded in gathering news from several Indonesian news outlets, although each news outlet employs different news syndication strategy. The system can also provide news articles recommendation according to user's feedback. Then, the time needed to gather news articles is increasing exponentially in line with the number of terms in articles.

REFERENCES

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems, 46,* 109-132. https://doi.org/10.1016/j.knosys.2013.03.012

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction, 12*(4), 331-370. https://doi.org/10.1023/A:1021240730564

Hameed, M. A., Jadaan, O. A., & Ramachandram, S. (2012). Collaborative filtering based recommendation system: A survey. *International Journal on Computer Science and Engineering, 4*(5), 859-876.

Karaa, W. B. A. (2013). A new stemmer to improve information retrieval. *International Journal of Network Security & Its Applications, 5*(4), 143-154. https://doi.org/10.5121/ijnsa.2013.5411

Palanivel, K., & Sivakumar, R. (2010). A study on implicit feedback in multicriteria e-commerce recommender system. *Journal of Electronic Commerce Research, 11*(2), 140-156.

Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web* (Vol. 4321, pp. 325-341). Springer Berlin Heidelberg.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing* (pp. 313-323). Englewood Cliffs, NJ: Prentice Hall Inc.

Ropero, J., Gómez, A., Carrasco, A., León, C., & Luque, J. (2012). Term weighting for information retrieval using fuzzy logic. In E. P. Dadios (Ed.), *Fuzzy logic-algorithms, techniques and implementations* (pp. 173-192). InTech.

Safoury, L., & Salah, A. (2013). Exploiting user demographic attributes for solving cold-start problem in recommender system. *Lecture Notes on Software Engineering, 1*(3), 303-307. https://doi.org/10.7763/LNSE.2013.V1.66

Said, A., Plumbaum, T., De Luca, E. W., & Albayrak, S. (2011). A comparison of how demographic data affects recommendation. In *19th International Conference on User Modeling, Adaptation and Personalization (UMAP)*.

Singh, A. K. (2012). Ajax asynchronous database refresh. *International Journal of Information and Communication Technology Research, 2*(8), 669-703.

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research, 37*, 141-188.

Zaman, A. N. K., Matsakis, P., & Brown, C. (2011). Evaluation of stop word lists in text retrieval using latent semantic indexing. In *The Sixth IEEE International Conference on Digital Information Management*. Melbourne, Australia.