Predicting User-specific Temporal Retweet Count

Bálint Daróczy[1] Róbert Pálovics[1,2] Vilmos Wieszner[3]
Richárd Farkas[3] András A. Benczúr[1]
[1]Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI)
[2]Technical University Budapest
[3]University of Szeged, Institute of Informatics
{daroczyb,rpalovics,benczur}@ilab.sztaki.hu, {wieszner,rfarkas}@inf.u-szeged.hu

ABSTRACT

Twitter generates a constant flow of quality news and mixed social content. While it is relative easy to separate large popularity news sources from personal messages, we address a more difficult question to predict the success of a single message among all messages of the same user. We describe a temporal evaluation framework to analyze which messages of which users will be retweeted the most. It turns out that global popularity depend mostly on the network characteristics of the user, while for a given user, the retweet count of a single message can be predicted best by using a variety of features of the content, including linguistic characteristics.

1. INTRODUCTION

Twitter, a mixture of a social network and a news media [21], has recently became the largest medium where users may spread information along their social contacts. Twitter users are a mix of quality news sources and "people-in-the-street" who generate a stream of very short, fragmentary stories of very different perspectives.

In this paper we investigate the temporal influence differences of the Twitter messages sent by the same user. Retweeting is a key act of highlighting the influence of a message [8]. By retweeting, users spreading information and build cascades of information pathways. Cha et al. [9] define influence as "... the power of capacity of causing an effect in indirect intangible ways...". In their key observation, the influence of a user is best characterized by the size of the audience who retweets rather than the size of the follower network. The distribution of retweet counts follows a power law [1].

Here, our objective is to predict the timely success of the information spread on the individual message level. We analyze how certain messages may reach out to a large number of Twitter users. In contrast to a similar investigation for analyzing the influence of users [3], we investigate each tweet by taking both the author user and the textual content of the message into account.

Our chief contribution is to find the difference between the popularity of a user and the success of a particular message among all tweets of the same user. We characterize the users both by the statistical properties of their follower network and their past retweet counts. The textual content is described by the terms of the normalized text and by several orthographic features along with deeper (psycho)linguistic ones that try to capture the modality of the message in question. While we use single content elements such as a given hashtag as well, consecutive bigrams and trigrams turn out to be the best performing predictors of cascade size.

Instead of focusing on either network or content only, we carried out an intensive feature engineering both at network and content analysis, and the added value of the two worlds was empirically evaluated. We defined a novel evaluation framework where we keep updating our prediction models and define a time aware evaluation. We compare classification and regression methods, including logistic regression, LogitBoost and different trees that we evaluate by AUC [13] for classification and among others RRSE (root relative squared error) for regression.

In our experiments we use the data set of [1] that consists of the messages and the corresponding user network of the Occupy movement. Our main findings can be summarized as

- High retweet counts can be predicted with particularly good accuracy immediately after the message appears.
- While the overall influence of a message depends on the popularity of the user, for a given user, the content and language determines how far the message will be retweeted.
- Among the most important language features, we find the level of uncertainty, hashtags and URLs. Bigrams and trigrams also play key role in prediction accuracy.
- Unlike in other results where logistic regression is used, we get significantly better performance by using Random Forest [12] for classifying the range of the cascade size and Regression Trees for predicting the size itself.

1.1 Related results

Social influence in Web based networks is investigated in several results: Bakshy et al. [5] model social contagion in the Second Life virtual world. Ghosh and Lerman [15] compares network measures for predicting the number of votes for Digg posts, who even give an empirical comparison of information contagion on Digg vs. Twitter [22]. In [16, 17], long discussion based cascades built from comments are investigated in four social networks, Slashdot (technology news), Barrapunto (Spanish Slashdot), Meneame (Span-

ish Digg) and Wikipedia. They propose models for cascade growth and estimate model parameters but give no size predictions.

From the **information spread** point of view, a number of related studies have largely descriptive focus, unlike our quantitative prediction goals. In [9] high correlation is observed between indegree, retweet and mention influence, while outdegree (the number of tweets sent by the user) is found to be heavily spammed. [21] reports similar findings on the relation among follower, mention and retweet influence. Several more results describe the specific means of information spread on Facebook [6, 2, 7].

There are only a limited number of related work on **retweet count prediction**. Cheng et al. [10] predict retweet count based on network features. Unlike in our result where we predict immediately after the tweet is published, they consider prediction after the first few retweets. The network features used in their work are similar to the ones in the present paper and in our earlier work [24]. The main contribution of this work is the investigation of content-based features and the interaction between network and content features. Petrovic et al. [26] predicts if a tweet will be retweeted at all, and give no evaluation on distinguishing between the messages of the same user. As another result very similar to the previous one, [20] give batch evaluation, for all users and the entire time range. They also use logistic regression; their features include tf.idf and an LDA based topic model. Similar to us, they classify for ranges of retweet counts, however they mention that their accuracy is very low for the mid-range. We include logistic regression among other classifiers as baseline methods in our work.

From the **content analysis** point of view, Bakshy et al. [3, 4] investigate `bit.ly` URLs but finds little connection between influence and URL content, unlike in our experiments where message content elements prove to be valuable for predicting influence. There has been several studies focusing exclusively on the analysis of the tweet message textual content to solve the re-tweet count prediction problem. Besides the terms of the message, Naveed et al. [23] introduced the features of direct message, mention, hashtag, URL, exclamation mark, question mark, positive and negative sentiment, positive and negative emoticons and valence, arousal, dominance lexicon features. Wang et al. [28] proposed deeper linguistic features like verb tense, named entities, discourse relations and sentence similarity. Similar to [26], neither of these results attempt to distinguish between the tweets of the same user.

Regarding the idea of **combining author, network and content information**, our work is related to Gupta et al. [18] who used these sources of information jointly for scoring tweets according to their credibility. Although credibility is related to social influence, the prediction of the credibility and the size of retweet cascade of a message requires different background information. Hence, we employ different network and content features.

2. DATA SET

The dataset was collected by Aragón et al. [1] using the Twitter API that we extended by a crawl of the user network. Our data set hence consists of two parts:

- *Tweet dataset:* tweet text and user metadata on the Occupy Wall Street movement[1].
- *Follower network:* The list of followers of users who posted at least one message in the tweet dataset.

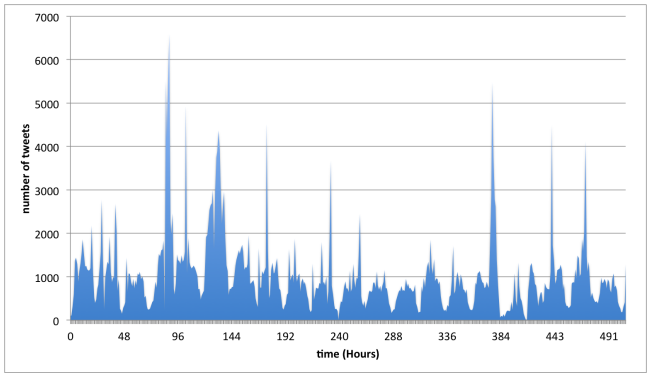


Figure 1: Temporal density of tweeting activity.

Table 1: Size of the tweet time series.

Number of users	371,401
Number of tweets	1,947,234
Number of retweets	1,272,443

Table 2: Size of the follower network.

Number of users	330,677
Number of edges	16,585,837
Average in/out degree	37

Table 1 shows the number of users and tweets in the dataset. One can see that a large part of the collected tweets are retweets. Table 2 contains the size of the crawled social networks. Note that the average in- and outdegree is relatively high. Fig. 1 shows the temporal density of tweeting activity.

For each tweet, our data contains

- tweet and user ID,
- timestamp of creation,
- hashtags used in the tweet, and
- the tweet text content.

In case of a retweet, we have all these information not only on the actual tweet, but also on the original *root tweet* that had been retweeted. We define the root tweet as the first occurrence of a given tweet.

3. RETWEET CASCADES

3.1 Constructing retweet cascades

In case of a retweet, the Twitter API provides us with the ID of the original tweet. By collecting retweets for a given original tweet ID, we may obtain the set users who have retweeted a given tweet with the corresponding retweet timestamps. The Twitter API however does not tell us the actual path of cascades if the original tweet was retweeted

[1]http://en.wikipedia.org/wiki/Occupy_Wall_Street

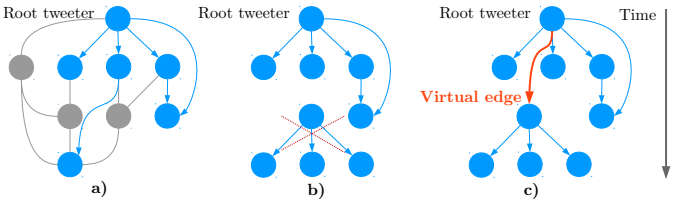


Figure 2: Creation of retweet cascades: Figure (a) shows the computation of the cascade edges. In Figures (b) and (c) we show the possible solutions in case of missing cascade edges.

Table 3: Examples of some highly retweeted messages in the data set.

message	retweet counts
@OWS_Live #OWS We can do the same reducing burning of fossil fuels too !!	325
Long Live The Peaceful Tea Party!! #gameon #college #twisters #ampat #sgp @OWS_Live #ows #violence #stupid #liberal #usefulidiots #getaclue	325
@[user] we need our own banking system by the people for the people. #OccupyWallStreet and have the 99% put their money there	319
The #NYPD officer who maced peaceful young women in the face got 10 vacation days docked. Not joking. [URL] #ows	143

several times. The information from the Twitter API on the tweet needs to be combined with the follower network to reconstruct the possible information pathways for a given tweet. However it can happen that for a given retweeter, more than one friend has retweeted the corresponding tweet before and hence we do not know the exact information source of the retweeter. The retweet ambiguity problem is well described in [3]. In what follows we consider all friends as possible information sources. In other words for a given tweet we consider all directed edges in the follower network in which information flow could occur (see Fig. 2 (a)).

3.2 Restoring missing cascade edges

For a given tweet, the computed edges define us a *retweet cascade*. However our dataset contains only a sample of tweets on the given hashtags and hence may not be complete: it can happen that a few intermediate retweeters are missing from our data. As a result, sometimes the reconstructed cascade graphs are disconnected. As detailed in Fig. 2 (b) and (c), we handle this problem in two different ways. One possible solution is to only consider the first connected component of the cascade (see Fig. 2 (b)). Another one is to connect each disconnected part to the root tweeter with one virtual cascade edge (see Fig. 2 (c)). In what follows, we work with cascades that contain virtual edges, therefore every retweeter is included in the cascade.

3.3 Examples of highly retweeted messages

In Table 3, we give a few examples of highly retweeted messages with the actual URLs and names replaced by [URL] and [name].

4. FEATURE ENGINEERING

To train our models, we generate features for each root tweet in the data and then we predict the future cascade size of the root tweet from these feature sets. For a given root tweet, we compute features about

- the author user and her follower network (*network features*) and
- the textual content of the tweet itself (*content features*).

Table 4 gives an overview of the feature templates used in our experiments.

4.1 Network Features

We consider statistics about the user and her cascades in the past as well as the influence and impressibility of her followers. We capture the influence and impressibility of a user from previously observed cascades by measuring the following quantities:

- *Number of tweets in different time frames:* for a given root tweet appeared in time t and a predefined time frame τ, we count the number of tweets generated by the corresponding user in the time interval $[t-\tau, t]$. We set τ for 1, 6, 12, 24, 48 and 168 hours.
- *Average number of tweets in different time frames:* We divide the number of tweets in a given time frame by τ.
- *User influence:* for a given user, we compute the number of times one of her followers retweeted her, divided by the number of the followers of the user.
- *User impressibility:* for a given user, we compute the number of times she retweeted one of her followees, divided by the number of followees of the user.

4.2 Content features

The first step of content processing is text normalization. We converted the text them into lower case form except those which are fully upper cased and replaced tokens by their stem given by the Porter stemming algorithm. We replaced user mentions (starting with '@') and numbers by placeholder strings and removed the punctuation marks.

The *content features* are extracted from the normalized texts. The basic feature template in text analysis consists the *terms* of the message. We used a simple whitespace tokenizer rather than a more sophisticated linguistic tokenizer as previous studies reported its empirical advantage [19]. We employed unigrams, bigrams and trigrams of tokens because longer phrases just hurt the performance of the system in our preliminary experiments.

Besides terms, we extracted the following features describing the *orthography* of the message:

- *Hashtags* are used to mark specific topics, they can be appended after the tweets or inline in the content, marked by #. From the counts of hashtags the user can tips the topic categories of tweet content but too many hashtag can be irritating to the readers as they just make confusion.

- *Telephone number:* If the tweet contains telephone number it is more likely to be spam or ads.
- *Urls:* The referred urls can navigate the reader to text, sound, and image information, like media elements and journals thus they can attract interested readers. We distinguish between full and truncated urls. The truncated urls are ended with three dot, its probably copied from other tweet content, so it was interested by somebody.
- The *like sign* is an illustrator, encouragement to others to share the tweet.
- The presence of a *question mark* indicates uncertainty. In Twitter, questions are usually rhetorical—people do not seek answers on Twitter [19]). The author more likely wants to make the reader think about the message content.
- The *Exclamation mark* highlights the part of the tweet, it expresses emotions and opinions.
- If *Numerical expressions* are present the facts are quantified then it is more likely to have real information content. The actual value of numbers were ignored.
- *Mentions:* If a user mentioned (referred) in the tweet the content of the tweet is probably connected to the mentioned user. It can have informal or private content.
- *Emoticons* are short character sequences representing emotions. We clustered the emoticons into positive, negative and neutral categories.

The last group of content features tries to capture the *modality* of the message:

- *Swear words* influence the style and attractiveness of the tweet. The reaction for swearing can be ignorance and also reattacking, which is not relevant in terms of retweet cascade size prediction. We extracted 458 swear words from `http://www.youswear.com`.
- *Weasel words and phrases*[2] aimed at creating an impression that a specific and/or meaningful statement has been made when in fact only a vague or ambiguous claim has been communicated. We used the weasel word lexicon of [27].
- We employed the linguistic inquiry categories (LIWC) [25] of the tweets' words as well. These categories describe words from emotional, cognitive and structural points of view. For example the "ask" word it is in Hear, Senses, Social and Present categories. Different LIWC categories can have different effect on the influence of the tweet in question.

4.3 N-grams

By using all the content features, we built n-grams as consecutive sequences in the tweet text that may include simply three terms ("posted a photo"), @-mentions, hashtags, URL ("@OccupyPics Photo http://t.co/..." coded as [[USER] Photo [URL]]), numbers ("has [NUMBER] followers"), non-alphanumeric ("right now !") as well as markers for swear or weasel expressions ("[WEASEL_WORD] people say"). We defined the following classes of n-grams, for $n \leq 3$:

[2]See `http://en.wikipedia.org/wiki/Wikipedia:Embrace_weasel_words`.

Table 4: Feature set.

network	*number of* {followers, tweets, root tweets}, *average* {cascade size, root cascade size}, *maximum* {cascade size, root cascade size}, *variance* of {cascade sizes, root cascade sizes}, *number of* tweets generated with different time frames, *time average* of the number of tweets in different time frames tweeter's influence and impressibility followers' average influence and impressibility
terms	normalized *unigrams, bigrams and trigrams*
orthographic	number of # with the values 0, 1, 2 ... 4 or 4 < number of {like *signs*, ?, !, mentions} number of full and truncated *urls* number of arabic *numbers* and *phone numbers* number of positive/negative/other *emoticons*
modality	number of swear words and weasel phrases union of the *inquiry categories* of the words

- **Modality**: The n-gram contains at least one swear or weasel word or expression (overall 208,368);
- **Orthographic**: No swear or weasel word but at least one orthographic term (overall 2,751,935);
- **Terms**: N-grams formed only of terms, no swear or weasel words and orthographic features (overall 771,196).

For efficiency, we selected the most frequent 1,000 n-grams from each class. The entire feature set hence consists of 3,000 trigrams.

5. TEMPORAL TRAINING AND EVALUATION

Here we describe the way we generate training and test sets for our algorithms detailed in Section 6. First, for each root tweet we compute the corresponding network and content features. We create daily re-trained models: for a given day t, we train a model on all root tweets that have been generated before t but appeared later than $t - \tau$, where τ is the preset time frame. After training based on the data before a given day, we compute our predictions for all root tweets appeared in that day.

In order to keep the features up to date, we recompute all network properties online, on the fly and use the new values to give predictions. By this method, we may immediately notice if a user starts gaining high attention or if a bursty event happens.

We take special attention to defining the values used for training and evaluation. For evaluation, we used the information till the end of the three week data set collection period, i.e. we used all the known tweets that belong to the given cascade. However, for training, we are only allowed to use and count the tweets up to the end of the training period. Since the testing period is longer, we linearly approximated the values for the remaining part of the testing period.

Our goal is to predict cascade size at the time when the root tweet is generated. One method we use is regression, which directly predict the size of the retweet cascade. For regression, we only use the global error measures:

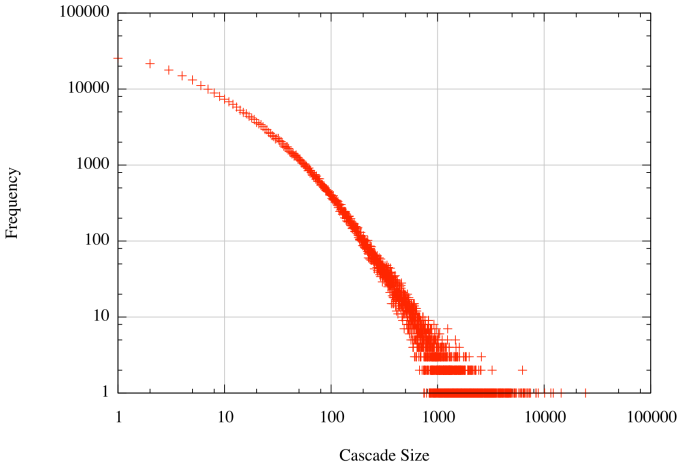


Figure 3: Cascade size distribution.

- Mean Average Error (MAE);
- Root Mean Squared Error (RMSE);
- Root Relative Squared Error (RRSE).

We also experiment with multiclass classification for ranges of the cascade size. The cascade size follows a power law distribution (see Fig. 3) and we defined three buckets, one with $0 \ldots 10$ (referred as "low"), one with $11 \ldots 100$ ("medium") and a largest one with more than 100 ("high") retweeters participating in the cascade. We evaluate performance by AUC [13] averaged for the three classes. Note that AUC has a probabilistic interpretation: for the example of the "high" class, the value of the AUC is equal to the probability that a random highly retweeted message is ranked before a random non-highly retweeted one.

By the probabilistic interpretation of AUC, we may realize that a classifier will perform well if it orders the users well with little consideration on their individual messages. Since our goal is to predict the messages in time and not the rather static user visibility and influence, we define new averaging schemes for predicting the success of individual messages.

We consider the classification of the messages of a single user and define two aggregations of the individual AUC values. First, we simply average the AUC values of users for each day (user average)

$$AUC_{\text{user}} = \frac{1}{N} \sum_{i=1}^{N} AUC_i, \tag{1}$$

Second, we are weighting the individual AUC values with the activity of the user (number of tweets by the user for the actual day)

$$AUC_{\text{wuser}} = \frac{\sum_{i=1}^{N} AUC_i T_i}{\sum_i^N T_i} \tag{2}$$

where T_i is the number of tweets by the i-th user.

We may also obtain regressors from the multiclass classification results. In order to make classification and regression comparable, we give a very simple transformation that replaces each class by a value that can be used as regressor. We select and use the training set average value in each class as the ideal value for the prediction.

6. RESULTS

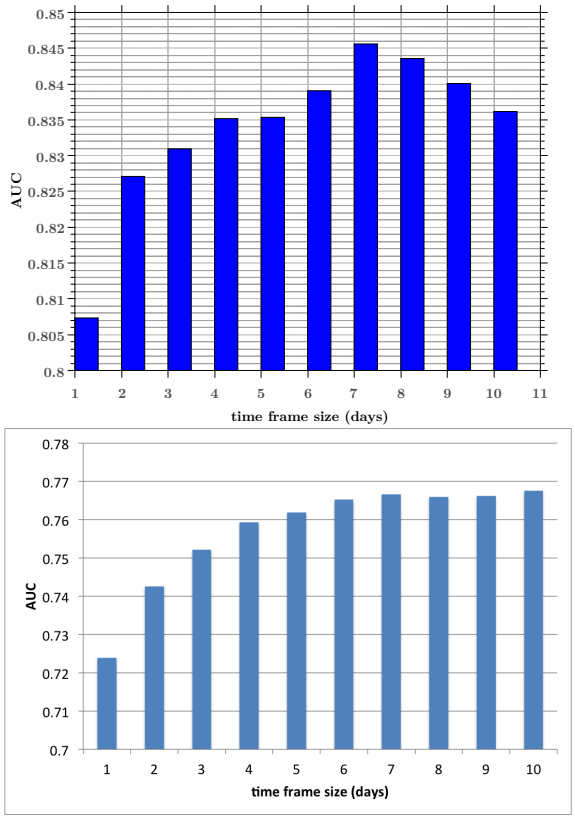


Figure 4: Daily average AUC of classifiers trained with different set of features, evaluated both as a global list (top) and as average on the user level by equation (1), bottom.

In this section, we train and evaluate first the classification and then the regression models to predict the future cascade size of tweets. We predict day by day, for each day in the testing period. For classification, we also evaluate on the user level by using equations (1) and (2). For classification, we show the best performing features as well.

As mentioned in Section 5, we may train our model with different τ. In Figure 4 we show the average AUC value with different time frames. As Twitter trends change rapidly, we achieve the best average results if we train our algorithms on root tweets that were generated in the previous week (approximately seven days), both for global and for user level average evaluation.

6.1 Cascade size by multiclass classification

First, we measure classifier performance by computing the average AUC values of the final results for the three size ranges. We were interested in how different classifiers perform and how different feature sets affect classifier performance. For this reason, we repeated our experiments with different feature subsets. Figure 5 shows our results. For each day, the network features give a strong baseline.

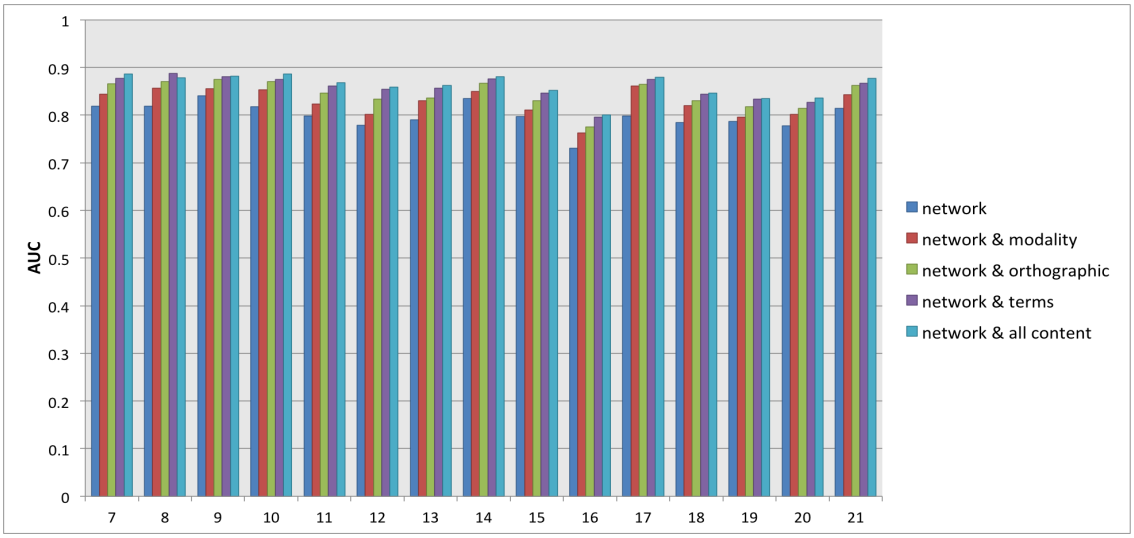


Figure 5: Daily average AUC of classifiers trained with different set of features.

Table 5: Retweet size classification daily average performance of different feature sets. The ideal values are MAE=2.435, RMSE=15.94, RRSE=0.414.

Features	Retweet range Low	Medium	High	Weighted Average	MAE	RMSE	RRSE
network	0.799	0.785	0.886	0.799	5.156	22.93	2.449
network & modality	0.827	0.814	0.905	0.827	4.843	22.40	2.033
network & orthographic	0.844	0.829	0.912	0.843	4.521	22.13	1.790
network & terms	0.857	0.847	0.914	0.857	4.157	21.90	1.323
network & all content	0.862	0.849	0.921	0.862	3.926	22.15	1.286

Table 6: Weighted average AUC over low, medium and high retweet range of different classifiers. Note that Multi-Layer Perceptron (MLP) did not terminate in 3 days for the large feature set.

Weighted Average AUC	network	network & all content
Random Forest	0.799	0.862
Logistic Regression	0.605	0.689
MLP	0.783	n/a

Table 7: Retweet size regression daily average performance of different feature sets.

Features	MAE	RMSE	RRSE
network, linear regression	3.225	14.30	0.909
network, MLP	3.015	14.91	0.716
network, RepTree	2.989	12.60	0.853
network & modality, RepTree	3.099	13.86	0.867
network & orthographic, RepTree	3.100	13.87	0.865
network & terms, RepTree	3.090	13.86	0.868
all, RepTree	3.100	13.87	0.865

The combination of these features with the content result in strong improvement in classifier performance. In Table 5 we summarize the average AUC values for different feature subsets over all four datasets. Our results are consistent: in all cases, the content related features improve the performance. Finally, we give the performance of other classifiers in Table 6 and conclude the superiority of the Random Forest classifier [12]. We use the classifier implementations of Weka [29] and LibLinear [11].

6.2 Cascade size by regression

We give regression results by the linear regression, multilayer perceptron and the regression tree implementation of Weka [29] in Table 7. As seen when compared to the last three columns in Table 5, regression methods outperform multiclass classification results transformed to regressors. Note that for the transformation, we use class averages obtained from the training data. If however we could perfectly classify the three classes, the ideal error values would be MAE=2.435, RMSE=15.94, RRSE=0.414. We could not reach close to the ideal values by regression either.

6.3 Cascade size on the user level

Our main evaluation is found in Table 8 where we consider the user level average AUC values as described in Section 5. As expected, since the new evaluation metrics give more emphasis on distinguishing between the tweets of the same user, we see even stronger gain of the modality and orthographic features.

6.4 Feature contribution analysis

We selected the most important network features by running a LogitBoost classifier [14]. The best features were all

Table 8: Retweet size classification daily average performance of different feature sets evaluated on the user level as defined in equations (1) and (2).

Retweet range		Low		Medium		High		Average	
Features		Uniform	Weighted	Uniform	Weighted	Uniform	Weighted	Uniform	Weighted
network	AUC	0.684	0.712	0.752	0.800	0.746	0.796	0.719	0.756
network & modality	AUC	0.700	0.722	0.751	0.796	0.737	0.756	0.726	0.757
network & orthographic	AUC	0.702	0.731	0.753	0.797	0.768	0.782	0.730	0.764
network & terms	AUC	0.705	0.732	0.757	0.800	0.767	0.786	0.733	0.766
network & all content	AUC	0.740	0.783	0.763	0.812	0.769	0.820	0.752	0.797

characterizing the network. We list the first five, in the order of importance:

1. The number of followers of the root tweet user;
2. The average cascade size of previous root tweets by the user.
3. The number of root tweets of the user so far (retweets excluded);
4. The average cascade size of previous tweets (including retweets) by the user;
5. The number of tweets of the user so far;

6.5 Content feature contribution analysis

We selected the most important content features by running logistic regression over the 3,000 trigrams described in Section 4.3. The features are complex expressions containing elements from the three major group of linguistic feature sets in the following order of absolute weight obtained by logistic regression:

1. Three words [marriage between democracy], in this order;
2. [at [HASHTAG_occupywallstreet][URL]]: the word "at", followed by the hashtag "#occupywallstreet", and a URL;
3. [between democracy and];
4. [capitalism is over];
5. [[HASHTAG_ows] pls];
6. [[WEASEL_WORD] marriage between]: the expression "marriage between" on the weasel word list, which counts as the third element of the trigram;
7. [[HASHTAG_zizek] at [HASHTAG_occupywallstreet]];
8. [[HASHTAG_occupywallstreet][URL][HASHTAG_auspol]];
9. [over [HASHTAG_zizek] at];
10. [calientan la]: means "heating up".

Note that all these features have negative weight for the upper two classes and positive or close to 0 for the lower class. Hence the appearance of these trigrams decrease the value obtained by the network feature based model. We may conclude that the use of weasel words and uninformative phrases reduce the chance of getting retweeted, as opposed to the sample highly retweeted messages in Table 3.

6.6 Frozen network features

To illustrate the importance of the temporal training and evaluation framework and the online update of the network features, we made an experiment where we replaced user features by static ones. The results are summarized in Table 9. Note that on the user level, all messages will have the same network features and hence classification will be random with AUC=0.5. In contrast, online updated network features are already capable of distinguishing between the messages of the same user, as seen in Tables 5 and 7.

Table 9: Retweet size classification with fixed user network features.

Features	Retweet range Low	Medium	High	Weighted Average
static network	0.798	0.779	0.868	0.797
static network & all content	0.854	0.804	0.932	0.851
static network per user	0.5	0.5	0.5	0.5
static network & all content per user	0.798	0.784	0.935	0.798

7. CONCLUSIONS

In this paper we investigated the possibility of predicting the future popularity of a recently appeared text message in Twitter's social networking system. Besides the typical user and network related features, we consider hashtag and linguistic analysis based ones as well. Our results do not only confirm the possibility of predicting the future popularity of a tweet, but also indicate that deep content analysis is important to improve the quality of the prediction.

In our experiments, we give high importance to the temporal aspects of the prediction: we predict immediately after the message is published, and we also evaluate on the user level. We consider user level evaluation key in temporal analysis, since the influence and popularity of a given user is relative stable while the retweet count of her particular messages may greatly vary in time. On the user level, we observe the importance of linguistic elements of the content.

Acknowledgments

We thank Andreas Kaltenbrunner for providing us with the Twitter data set [1].

8. REFERENCES

[1] P. Aragón, K. E. Kappler, A. Kaltenbrunner, D. Laniado, and Y. Volkovich. Communication dynamics in twitter during political campaigns: The case of the 2011 spanish national election. *Policy & Internet*, 5(2):183–206, 2013.

[2] E. Bakshy, D. Eckles, R. Yan, and I. Rosenn. Social influence in social advertising: evidence from field experiments. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 146–161. ACM, 2012.

[3] E. Bakshy, J. M. H., W. A. Mason, and D. J. Watts. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 65–74. ACM, 2011.

[4] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. Identifying influencers on twitter. In *Fourth ACM International Conference on Web Seach and Data Mining (WSDM)*, 2011.

[5] E. Bakshy, B. Karrer, and L. A. Adamic. Social influence and the diffusion of user-created content. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 325–334. ACM, 2009.

[6] E. Bakshy, I. Rosenn, C. Marlow, and L. Adamic. The role of social networks in information diffusion. In *Proceedings of the 21st international conference on World Wide Web*, pages 519–528. ACM, 2012.

[7] M. S. Bernstein, E. Bakshy, M. Burke, and B. Karrer. Quantifying the invisible audience in social networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 21–30. ACM, 2013.

[8] D. Boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–10. IEEE, 2010.

[9] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *4th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010.

[10] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec. Can cascades be predicted? In *Proceedings of the 23rd international conference on World wide web*, pages 925–936. International World Wide Web Conferences Steering Committee, 2014.

[11] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[12] FastRandomForest. Re-implementation of the random forest classifier for the weka environment. `http://code.google.com/p/fast-random-forest/`.

[13] J. Fogarty, R. S. Baker, and S. E. Hudson. Case studies in the use of roc curve analysis for sensor-based estimates in human computer interaction. In *Proceedings of Graphics Interface 2005*, GI '05, pages 129–136, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society.

[14] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of statistics*, pages 337–374, 2000.

[15] R. Ghosh and K. Lerman. Predicting influential users in online social networks. *arXiv preprint arXiv:1005.4882*, 2010.

[16] V. Gómez, H. J. Kappen, and A. Kaltenbrunner. Modeling the structure and evolution of discussion cascades. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, pages 181–190. ACM, 2011.

[17] V. Gómez, H. J. Kappen, N. Litvak, and A. Kaltenbrunner. A likelihood-based framework for the analysis of discussion threads. *World Wide Web*, pages 1–31, 2012.

[18] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier. Tweetcred: Real-time credibility assessment of content on twitter. In *Social Informatics*, volume 8851 of *Lecture Notes in Computer Science*, pages 228–243. 2014.

[19] V. Hangya and R. Farkas. Filtering and polarity detection for reputation management on tweets. In *Working Notes of CLEF 2013 Evaluation Labs and Workshop*, 2013.

[20] L. Hong, O. Dan, and B. D. Davison. Predicting popular messages in twitter. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 57–58, New York, NY, USA, 2011. ACM.

[21] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[22] K. Lerman and R. Ghosh. Information contagion: An empirical study of the spread of news on digg and twitter social networks. In *Proceedings of 4th International Conference on Weblogs and Social Media (ICWSM)*, 2010.

[23] N. Naveed, T. Gottron, J. Kunegis, and A. C. Alhadi. Bad news travel fast: A content-based analysis of interestingness on twitter. In *Proceedings of the 3rd International Web Science Conference*, WebSci '11. ACM, 2011.

[24] R. Palovics, B. Daroczy, and A. Benczur. Temporal prediction of retweet count. In *Cognitive Infocommunications (CogInfoCom), 2013 IEEE 4th International Conference on*, pages 267–270. IEEE, 2013.

[25] J. Pennebaker, C. Chung, M. Ireland, A. Gonzales, and R. Booth. The development and psychometric properties of liwc2007. Technical report, University of Texas at Austin, 2007.

[26] S. Petrovic, M. Osborne, and V. Lavrenko. Rt to win! predicting message propagation in twitter. In *ICWSM*, 2011.

[27] Gy. Szarvas, V. Vincze, R. Farkas, Gy. Móra, and I. Gurevych. Cross-genre and cross-domain detection of semantic uncertainty. *Computational Linguistics*, 38(2):335–367, 2012.

[28] A. Wang, T. Chen, and M.-Y. Kan. Re-tweeting from a linguistic perspective. In *Proceedings of the Second Workshop on Language in Social Media*, pages 46–55, 2012.

[29] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, second edition, June 2005.

Combining Collaborative Filtering and Search Engine into Hybrid News Recommendations

Toon De Pessemier
iMinds-Ghent University
G. Crommenlaan 8 / 201
B-9050 Ghent, Belgium
toon.depessemier@ugent.be

Sam Leroux
iMinds-Ghent University
G. Crommenlaan 8 / 201
B-9050 Ghent, Belgium
sam.leroux@ugent.be

Kris Vanhecke
iMinds-Ghent University
G. Crommenlaan 8 / 201
B-9050 Ghent, Belgium
kris.vanhecke@ugent.be

Luc Martens
iMinds-Ghent University
G. Crommenlaan 8 / 201
B-9050 Ghent, Belgium
luc1.martens@ugent.be

ABSTRACT

Recommender systems have proven their usefulness in many classical domains, such as movies, books, and music, in helping users to overcome the information overload problem. When properly configured, recommender systems can also act as a supporting tool for content selection and retrieval in more challenging fields, such as news content. The short life span of news items and the demand for up-to-date recommendations require a specially tailored approach. This paper proposes a hybrid recommender system using a search engine as a content-based approach and combining this with collaborative filtering for diversifying the user profiles. Based on similar users, user profile vectors are extended with related terms interesting to read about. The recommender system is fed real-time streams of news content originating from different sources. The resulting recommendations are clustered into topics and presented through a web application. This paper demonstrates that the advantages of both search engine and collaborative filtering can be successfully combined into a recommender system for domains with transient items, such as news.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering; H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Recommender system, Hybrid, Real-time, News, Content-based, Storm, Collaborative filtering

1. INTRODUCTION

Recommender systems are software tools and techniques providing suggestions for items that may be of interest to a user such as videos, songs, or the products of an online shop. Although most research on recommender systems has been performed in these traditional content domains, recommender systems have also been deployed for services that focus on more transient items, characterized by a short life span. Cultural events, such as concerts or theater performances, are only available during a certain time period, in which the recommender has to learn users' preferences for these events as well as generating recommendations for interested users [8]. Reciprocal recommender systems recommend people to people such as for dating, employment, and mentoring services. Recommendations are only successful if both people like each other. However, when people are successful in finding a date, a job, or a mentor, they may never return to the web site [19], thereby limiting the availability in time of candidate recommendations. For news content, items quickly lose their information value and should therefore be recommended as soon as they are available in order to minimize delay between production and consumption of the content. A preview of a sports game has lost any information value after the game for instance. Especially for online news, fast delivering and recommending of content is of utmost importance.

For content with a short life span, and for news content in particular, collaborative filtering (CF) systems have difficulties to generate recommendations because of the new item problem (cfr. cold start problem). CF requires a critical amount of consumptions (explicit or implicit feedback) before an item can be recommended. Once enough consumption data is available, the information value of the content might be degraded, making recommendations for the content useless. Therefore, content-based or hybrid approaches are considered as more suitable for news recommendation [9].

2. RELATED WORK

In the domain of digital news services, various initiatives to personalize the offered news content have been proposed. One of the first recommender systems for personalizing news

content was GroupLens [14]. GroupLens used collaborative filtering to generate recommendations for Usenet news and was evaluated by a public trial with users from over a dozen newsgroups. This research identified some important challenges involved in creating a news recommender system.

SCENE [15] is such a news service. It stands for a SCalable two-stage pErsonalized News rEcommendation system. The system considers characteristics such as news content, access patterns, named entities, popularity, and recency of news items when performing recommendation. The proposed news selection mechanism demonstrates the importance of a good balance between user interests, the novelty, and diversity of the recommendations.

The News@hand system [5] is a news recommender which applies semantic-based technologies to describe and relate news contents and user preferences in order to produce enhanced recommendations. This news system ensures multimedia source applicability. The resultant recommendations can be adapted to the current context of interest, thereby emphasizing the importance of contextualization in the domain of news.

In the CLEF NEWSREEL track [3], news recommendation techniques could be evaluated in real-time by providing news recommendations to actual users that visit commercial news portals. A web-based platform is used to distribute recommendations to the users and return users' impressions of the recommendations to the researchers.

The News Recommender Systems Challenge [22] focused on providing live recommendations for readers of German news media articles. This challenge highlighted why news recommendations have not been analyzed as thoroughly as some of the other domains such as movies, books, or music. Reasons for this include the lack of data sets as well as the lack of open systems to deploy algorithms in. In the challenge, the deployed recommenders for generating news recommendations are: Recent Recommender (based only on the recency of the articles), Lucene Recommender (a text retrieval system built on top of Apache Lucene), Category-based Recommender (using the article's category), User Filter (filters out the articles previously observed by the current user), and Combined Recommender (a stack or cascade of two or more of the above recommenders).

The usefulness of retrieval algorithms for content-based recommendations has been demonstrated with experiments using a large data set of news content [2]. Binary and graded evaluation were compared and graded evaluation showed to be intrinsically better for news recommendations. This study emphasizes the potential of combining content-based approaches with collaborative filtering into a hybrid recommender system for news.

Although the various initiatives emphasize the importance of a personalized news offer, most of them focus on the recommendation algorithms. However, the way in which content is gathered, delivered, and presented to end-users is of crucial importance for a successful service. Users want an up-to-date, personalized news offer, providing a complete overview of all news events, which is clearly structured and classified by topic. In this study, the focus is not on improving state of the art recommendation algorithms or search engines, since many studies covered this already [22, 3, 6, 2]. The focus of this paper is rather on investigating the real-time aspect of delivering personalized recommendations (up-to-date content offer), the aggregation of multiple content sources of a different nature, such as premium content, blogs, Twitter, etc. (complete overview), and the clustering of content items by topic (clearly structured).

The remainder of this paper is structured as follows. Section 3 compares the recommendation and content retrieval problem and indicates resemblances between the two approaches. Section 4 discusses the architecture of our system and zooms in on the data fetching, search engine, recommender, and clustering component of the proposed system. Section 5 provides details on the implementation, the user interaction with the system, and the user interface. Finally, Section 6 draws conclusions.

3. RECOMMENDATION AS A CONTENT RETRIEVAL PROBLEM

Content-based algorithms typically compare a representation of the user profile with (the metadata of) the content, and deliver the best matching items as recommendations [16]. These algorithms often use relatively simple retrieval models, such as keyword matching or the Vector Space Model (VSM) with basic Term Frequency - Inverse Document Frequency (TF-IDF) weighting [17]. As such, the matching process of content and profile in a content-based algorithm shows many resemblances with the content retrieval process of a search engine.

Before employing the VSM and TF-IDF weighting in a content-based algorithm, preprocessing of the content is often required. If the content consists of complete sentences, the text stream must be broken up into tokens: phrases, words, symbols or other meaningful elements. Tokens that belong together, e.g. United States of America or New York, deserve special attention, and can be handled by reasoning based on uppercase letters and n-gram models [4]. Before further processing of the content, the next operation is filtering out stop words, the most common words in a language that typically have a limited intrinsic value. Another important operation is stemming, the process for reducing inflected (or sometimes derived) words to their word stem, or root form. In our implementation, Snowball [20] is used, a powerful stemmer for the English language. Again, a resemblance with content retrieval processes can be noticed, since these preprocessing operations are also performed during the indexing of web pages in search engines.

Based on these similarities between the content recommendation and content retrieval problem, we opted to utilize a search engine as the core component of our recommender service. The user profile is used as search query and provides the input for the search engine. Consequently, the search results are the content items best matching the user profile and can therefore be considered as personalized recommendations for the user.

Utilizing a search engine to generate personalized recommendations for news content brings some additional advantages.

- *Short response time.* Search engines are strongly optimized to quickly identify and retrieve relevant content items. An inverted index [6] is used as a very efficient structuring of the content, enabling to handle massive amounts of documents.
- *Fast processing of new content.* New content items can be processed quickly by making additions to the index structure, thereby making these new content items

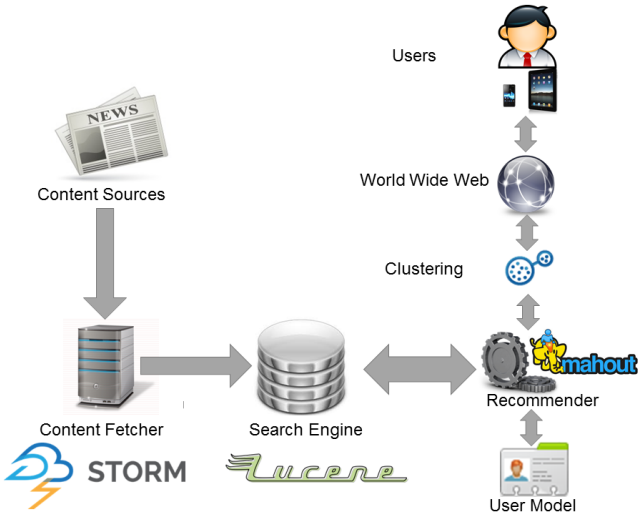


Figure 1: The architecture and content flow of the news recommender system.

available for recommendation almost immediately. In contrast, traditional recommender systems often require intensive calculations of similarities before a new item can be recommended.

- *Limited storage requirements.* The index structure of search engines is a very efficient storage way to retrieve documents.

4. ARCHITECTURE

Figure 1 shows the architecture and content flow of the news recommender system. The different components will be discussed in more detail in this section.

4.1 Data Fetching

The first phase of recommendation process is to *fetch the news content* periodically from different sources. When new items are available, their content is fetched and processed. Many online news services provide their content through RSS-feeds. To parse these feeds, the Rome project [28] is used since this is a robust parser. Besides RSS-feeds, other sources, such as blogs, can also be incorporated into the system by using a specific content parser.

In order to keep track of the most recent news content, news sources are checked regularly for new content. Different news sources have a different publishing frequency, ranging from one news item per day, to multiple news items per minute. Therefore, we used a simple mechanism to adapt the frequency of checking for new content to the publishing frequency of the content source. For each content source, a dynamic timer is used to determine when to check for new content. After a timeout, the content is fetched. If new content is available, the content item is added to the search engine and the timeout is reduced by half. If no new content is available, the timeout is doubled. This simple mechanism showed to be sufficient as a convergence method for the timeout parameter.

In order to process the stream of incoming news articles of different sources continuously, Apache Storm [1] was used. Storm enables the processing of large streams of data in real time. As opposed to batch processing, Storm handles the news articles as soon as these are available. To use Storm, a topology composed of 'Spouts' and 'Bolts' has to be built, which describes how messages flow into the system and how they have to be processed. A Spout is a source of data streams. A Bolt consumes any number of data streams, does some processing, and can emit new data streams. Storm can make duplicates of these components, and even distribute these duplicates over multiple machines, in order to process large amounts of data. As a result, Storm makes the system scalable and distributed.

In our implementation, the Spouts input data into the system as URLs of RSS-feeds, blogs, or social network accounts. Storm will distribute the work load over different Bolts of the first type, which fetch the data from the feeds. In case new articles are available in the feed, the URL of these articles is passed to the Bolts of the second type. These Bolts fetch the article content and remove non-topical information, such as advertisements, by identifying specific HTML tags in the source code of the web page. Subsequently, the Bolts pass the article content to Bolts of the third type. The task of Bolts of the third type is to analyze the content and obtain information such as the title, date, category, etc. Next, the article content is passed to the fourth type of Bolts, which will input the news articles into the search engine. After inputting the content into the search engine, statistical information about the article content is stored by the fifth and last type of Bolts. E.g., the frequency of occurrence of a term at a specific moment in time is used to determine if a news topic is trending and important (Section 4.3).

4.2 Search Engine

In the second phase, the content is processed by a *search engine*. We opted to use Apache Lucene [24], a Java library that is typically used for services handling large amounts of data and offering search functionalities. Since Lucene's performance, simplicity, and ease-of-use have been investigated in related work [12], this research does not focus on the characteristics of Lucene, but rather on the combination of search engine and recommender system.

As alternative search engines, we considered Solr [26] and ElasticSearch [10]. Solr is a ready-to-use, open source search engine based on Lucene. In comparison with Lucene, Solr provides more specific features such as a REST webinterface to index and search for documents. However, the disadvantage of Solr is that some of the specialized functionality is hidden and not directly usable. Besides, the overhead of the webinterface of Solr introduced some delay in comparison with Lucene in our experiments. Similar to Solr, ElasticSearch hides some of Lucene's functionality by using a simple web interface. Specific information about the content items, such as the term frequencies or statistics about the complete index, are not directly accessible using ElasticSearch. Therefore, Lucene was chosen to provide the functionality of the search engine. In case the processing load for the Lucene index becomes an issue, distribution over different machines is possible by solutions such as Katta [13], thereby making it scalable.

4.3 Recommender

In the third phase, personalized recommendations are generated. The user profile is used as a search query and sent to the search engine. The resulting search results are consid-

ered as personalized recommendations. As is common practice in the VSM [16], the user profile is modeled as a vector of terms (tags) together with a value specifying the user's interest in the term. These terms are words (or N-grams) in the article that are identified as relevant for the content. The current implementation is based on the traditional TF-IDF, but alternative solutions can easily be integrated. When the user reads a news article, the profile vector is updated with the TF-IDF values of the terms of the article. However, this update process is only executed if the user has spent more time on the article than a predefined threshold. In our implementation, we have chosen 10 seconds as a minimum time period for users to read the title and get an impression of the article content. More advanced approaches are possible using the reading time and article length, but these are not always reliable in a mobile environment.

Since our system uses implicit feedback based on users' selections (see Section 5), the profile update process is a simple summation of the item vectors of different articles. Articles from the past are considered as less representative for the user's preferences than recent articles. Therefore, the value of a term decreases exponentially as the age (in hours) of the article increases, meaning that older items will contribute less to the profile. Although these terms with their corresponding interest values may form a rather long profile vector, and as a result a long search query, Lucene is designed to handle such search requests in a very short time. Therefore, recommendations are requested when needed and hence always up-to-date.

News events with a high impact (e.g., a natural disaster in a remote part of the world) have to be detected and considered as a recommendation, even if the topic does not completely match the user's interests. These *trending topics* can be identified based on their frequency of occurrence. If the current frequency of occurrence is significantly higher than the frequency of occurrence in the past, the topic is considered as trending. Besides, trending topics are discovered by checking trends on Google's search queries [11]. Every hour, Google publishes a short list with trending searches. A special Spout was implemented to fetch these trending topics hourly. Trending topics are used to create a query for the search engine, and the resulting news items are added to the user's recommendation list. A final source of trending topics is Twitter. Research has shown that Twitter messages are a good reflection of topical news [18]. Therefore, another Spout was assigned specifically to query tweets regarding news topics using the Twitter API. Twitter accounts of specialized news services and newspapers were followed. The tweets originating from these accounts are focusing on recent news and characterized by a high quality. Retweets and Favorites give an indication of the popularity and impact of a tweet. Subsequently, Tweets are processed in the same manner as other news items by Bolts.

As stated in the introduction, straightforward collaborative filtering is not usable for news recommendations because of the new item problem. Unfortunately, content-based recommendations are typically characterized by a low serendipity; recommendations are too obvious. To introduce serendipity, a hybrid approach was taken by adding a collaborative filtering aspect to the content based recommender. A traditional nearest neighbor approach was used to calculate similarities between user-user pairs. Instead of recommending the items that the neighbors have consumed, our implementation will recommend profile terms that are prominent in neighboring profiles. These profile terms of the neighbors are used to extend the profile of the user, thereby making it more diverse. Subsequently, this extended profile is used to generate content-based recommendations using the search engine. By extending the profile of a user with terms that are significant in the profiles of the user's neighbors, profiles are broadened and diversified with related terms. These extended profiles will produce more diverse recommendations covering a broad range of topics. Since the additional profile terms are originating from neighbors' profiles, the added terms will probably be in the area of interest of the user. The collaborative filtering component is based on the implementation of Apache Mahout [25]. Mahout ensures the scalability of this component of the system. Moreover, the profile extension is not a time-critical component, and is therefore implemented as a batch process running periodically. Content-based recommendations are based on the current version of the user profile, and as soon as the profile extension is finished, the profile is updated. This ensures that real-time recommendations can be generated at all time.

Finally, also the publishing date of the article is taken into account in the recommendation process. In the current implementation, only news articles of the last two days are candidate recommendations. However, a more intelligent degradation over time, with a degradation rate depending on the category or content of the article, can be future work.

4.4 Clustering

In the fourth phase, the recommended news items are clustered into topics. Since the news items in our system originate from different content sources, multiple items may cover the same news story. To provide users a clear overview of the news without removing content items, items about the same topic are clustered together. To cluster the content, three clustering approaches are considered during the design.

1. A *periodic clustering* of the complete content library before generating recommendations. Traditional clustering algorithms, which assume that all items are known before the clustering starts, can be used to periodically cluster all news items [23]. This approach does not allow the recommendation process to begin before the complete clustering of the content library is finished. Since this disadvantage introduces too much delay when adding new content to the library, it was not an approach for our system.

2. An *incremental clustering* of the content library before generating recommendations. In this approach, new content items are assigned to the best matching cluster, or a new cluster is made in case there is no match. Although this clustering approach is used in different existing systems [15, 7], we did not opt for this approach because it is not personalized. For a large content library, a large number of clusters can be identified. Since the clustering process is performed before the recommendation process, the clusters are identical for all users. However, personal interests may require a personalized clustering of the news content.

3. A *clustering of the recommended content items.* This is the approach that is used in our system, using a hierarchical clustering algorithm. Content items are not

clustered until the recommendation process is finished. The advantage of this approach is that only a small set of content items (250 candidate recommendations in our system) have to be clustered. Another advantage of clustering the recommendation results is the personalized nature of this set. For each user, the clustering process will result in a different clustering. Even a different level of clustering (number of clusters) can be chosen for every user. Users who are very interested in sports may find different clusters for soccer, baseball, cycling, etc., whereas users who are moderately interested may receive only one sports cluster containing all sporting disciplines. On the downside, users may not be familiar with a personalized clustering. As user preferences change or as collaborative filtering is applied to extend profile vectors, clusters are not stable over time. This behavior may surprise users who first got used to the existing clusters and then cannot find their 'old favorite' clusters anymore.

5. USER INTERACTION

Mobile has become, especially amongst younger media consumers, the first gateway to most news events published online. In a recent survey [21], conducted in 10 countries with high Internet penetration, one-fifth of the users now claim that their mobile phone is the primary access point for news. The small screen and typical interaction methods of mobile devices (touch screen) induce extra challenges and possibilities for news services.

Because of this, we made our news service available as a web application that is usable on desktop but also on tablets and smartphones. Figure 2 shows a screenshot of the user interface of the (mobile) web application, based on HTML5 and Javascript. On the left hand side, an overview of the recommended content items is shown. For each article, the number indicates how many articles covering this topic are clustered together. Selecting one of the items in the left column will show the article content on the right hand side using an HTML iframe. HTML iframes are used in order to provide all functionality of the source website, such as hyperlinks, while providing users the ability to browse their recommendations using the left column. Parsing the content of the source and reproducing it inside our own application is a technically feasible alternative, but violates the terms of use of many websites. Redirecting the users to the source website (using hyperlinks) would imply that users leave our web application and continue their news consumption on the source website, thereby making it impossible to track their behavior. The user interface is adapted to mobile devices by providing a clearly readable overview of the content, and interaction through tapping and swiping the touch screen. For smaller screens, such as smartphones, the column on the left hand side can be hidden to show the news articles in full screen. Further optimizations for mobile devices and touch screens are provided by using JQuery Mobile [27].

Explicit feedback for news services is difficult to interpret and therefore less common. E.g., a 1-star on a 5 point rating scale can be interpreted as a disinterest for the content, or as sympathizing with a story about some tragic event. Therefore, our system is using implicit feedback based on the user's viewing behavior. If an article is selected and shown on the screen for at least 10 seconds, we assume that the user has some interest in the topic of the story .

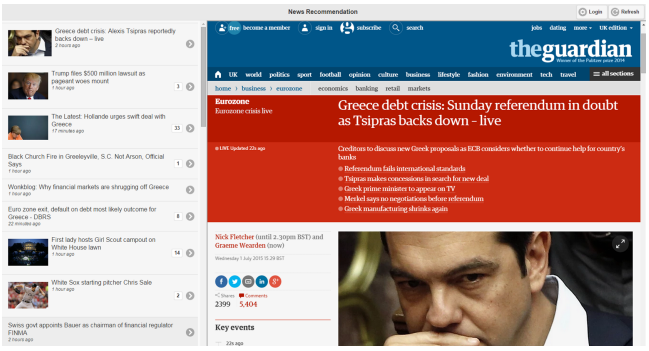


Figure 2: A screenshot of the user interface of the (mobile) web application.

Evaluating the system performance in terms of response time gave the following results. A mean response time of 800 ms was measured to generate 250 recommendations. This request includes retrieving the user profile and trending terms, executing the query on the search engine, and clustering the resulting items. These results were obtained on our test system, an Intel Xeon E5645 CPU at 2.40 GHz with 8GB of RAM running CentOS 6.6.

6. CONCLUSIONS

In this paper, we proposed a hybrid, real-time recommender system for news, combining technologies such as Storm, Lucene, and Mahout to ensure scalability and quick response times. Storm enables the processing of large streams of news content. Lucene provides the functionality of a search engine and is used as a content-based recommender. The collaborative filter of Mahout is used to exchange profile terms among neighboring users. User profile vectors are extended with related terms interesting to read about. The resulting hybrid recommendations are clustered according to their topic and presented to the user through a web application that is optimized for mobile devices. This research discussed the possibility of combining collaborative filtering and a search engine to compose a hybrid news recommender system, thereby combining the advantages of both. Search engines ensure a real-time response behavior while collaborative filtering adds community knowledge to the system. As future work, we consider to make a distinction between short-term interests and long-term interests of users. We also plan to focus more on entities mentioned in articles.

7. ACKNOWLEDGMENTS

We would like to thank Sam Leroux for the work he performed in the context of this research during his master thesis.

8. REFERENCES

[1] Apache Software Foundation. Apache storm, 2015. Available at `http://storm.apache.org/`.

[2] T. Bogers and A. van den Bosch. Comparing and evaluating information retrieval algorithms for news recommendation. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, pages 141–144, New York, NY, USA, 2007. ACM.

[3] T. Brodt and F. Hopfgartner. Shedding light on a living lab: The clef newsreel open recommendation platform. In *Proceedings of the 5th Information Interaction in Context Symposium*, IIiX '14, pages 223–226, New York, NY, USA, 2014. ACM.

[4] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, Dec. 1992.

[5] I. Cantador, A. Bellogín, and P. Castells. News@hand: A semantic web approach to recommending news. In W. Nejdl, J. Kay, P. Pu, and E. Herder, editors, *Adaptive Hypermedia and Adaptive Web-Based Systems*, volume 5149 of *Lecture Notes in Computer Science*, pages 279–283. Springer Berlin Heidelberg, 2008.

[6] D. Cutting and J. Pedersen. Optimization for dynamic inverted index maintenance. In *Proceedings of the 13th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '90, pages 405–411, New York, NY, USA, 1990. ACM.

[7] A. S. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: Scalable online collaborative filtering. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 271–280, New York, NY, USA, 2007. ACM.

[8] T. De Pessemier, S. Coppens, K. Geebelen, C. Vleugels, S. Bannier, E. Mannens, K. Vanhecke, and L. Martens. Collaborative recommendations with content-based filters for cultural activities via a scalable event distribution platform. *Multimedia Tools and Applications*, 58(1):167–213, 2012.

[9] T. De Pessemier, C. Courtois, K. Vanhecke, K. Van Damme, L. Martens, and L. De Marez. A user-centric evaluation of context-aware recommendations for a mobile news service. *Multimedia Tools and Applications*, pages 1–29, 2015.

[10] Elastic. Elasticsearch, 2015. Available at `https://www.elastic.co/`.

[11] Google. Google Hourly Trends, 2015. Available at `http://www.google.com/trends/hottrends/atom/hourly`.

[12] E. Hatcher and O. Gospodnetic. Lucene in action (in action series). 2004.

[13] Katta. Lucune & more in the cloud, 2015. Available at `http://katta.sourceforge.net/`.

[14] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to usenet news. *Commun. ACM*, 40(3):77–87, Mar. 1997.

[15] L. Li, D. Wang, T. Li, D. Knox, and B. Padmanabhan. Scene: A scalable two-stage personalized news recommendation system. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 125–134, New York, NY, USA, 2011. ACM.

[16] P. Lops, M. de Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 73–105. Springer US, 2011.

[17] C. D. Manning, P. Raghavan, H. Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[18] O. Phelan, K. McCarthy, and B. Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 385–388, New York, NY, USA, 2009. ACM.

[19] L. Pizzato, T. Rej, T. Chung, I. Koprinska, and J. Kay. Recon: A reciprocal recommender for online dating. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 207–214, New York, NY, USA, 2010. ACM.

[20] M. F. Porter. Snowball: A language for stemming algorithms, 2001. Available at `http://snowball.tartarus.org/`.

[21] Reuters Institute for the Study of Journalism. Digital News Report, 2015. Available at `http://www.digitalnewsreport.org/`.

[22] A. Said, A. Bellogín, and A. de Vries. News recommendation in the wild: Cwi's recommendation algorithms in the NRS challenge. In *Proceedings of the 2013 International News Recommender Systems Workshop and Challenge. NRS*, volume 13, 2013.

[23] K. G. Saranya and G. S. Sadhasivam. A personalized online news recommendation system. *International Journal of Computer Applications*, 57(18):6–14, November 2012.

[24] The Apache Software Foundation. Apache Lucene, 2015. Available at `https://lucene.apache.org/`.

[25] The Apache Software Foundation. Apache Mahout, 2015. Available at `http://mahout.apache.org/users/recommender/recommender-documentation.html`.

[26] The Apache Software Foundation. Apache Solr, 2015. Available at `http://lucene.apache.org/solr/`.

[27] The jQuery Foundation. jQuery mobile, a touch-optimized web framework, 2015. Available at `http://jquerymobile.com`.

[28] M. Woodman. Rome, 2015. Available at `https://rometools.jira.com/wiki/display/ROME/Home`.

Survey of User Profiling in News Recommender Systems

Mahboobeh Harandi
School of Information Studies
Syracuse University
Syracuse, USA
mahboobh@stud.ntnu.no

Jon Atle Gulla
Department of Computer and Information Science, NTNU
Trondheim, Norway
jag@idi.ntnu.no

ABSTRACT

In order to personalize news articles in online news recommender systems, a number of user profiling techniques have been employed. Both long-term and short-term interests of the user are captured in this process and are important to the construction of user profiles. Due to the short life span and the unstructured format of news articles, the changing interests of the user and the lack of explicit feedback, user profiling is challenging in the news domain. Natural language processing and machine learning techniques are used, though there is no accepted best approach to user profiling. In this survey we discuss the most common user profiling techniques in news recommendation and show how they can be classified according to features used and challenges addressed.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous

Keywords

Recommender systems, user profiling, supervised and unsupervised learning techniques, natural language processing, content-based filtering, collaborative filtering, hybrid filtering, news recommendation

1. INTRODUCTION

Recommender systems have become popular over the last few years because they have the ability to provide relevant information to users based on their needs or interests. Their goal is to filter out information and present only the required and interesting pieces to the user. This type of application need to understand the information they manage, but also to understand the users' behavior and their underlying information needs.

With the abundance of information on the web, there is a problem that users get overwhelmed while looking for the desired information. The users cannot be asked to browse through hundreds of items to find the correct one, or to formulate unambiguous search queries that can pick out the correct document from millions of similar documents. Recommender systems try to analyze previous user behavior to understand what the user might find interesting in the future. There are today several popular recommender systems used in a wide variety of domains. In the movie domain, Netflix is a successful pioneer of the technology and can on-the-fly recommend movies to users on the basis of what s/he has watched and what s/he has rated. The company has more than 50 million subscribers all around the world and has been fairly successful in monetizing the technology and coming up with attractive personalized movie recommendations. Another successful application of recommender systems is found in Amazon, where different items are proposed to the user based on their shopping history.

News recommender systems are another popular application of recommendation technologies. These systems provide interesting and integrated news from thousands of news stories from media houses and news agencies. From the user perspective, it is helpful if the recommender system can propose interesting articles instead of forcing the reader to spend too much time looking for them. Times, Google News, Daily learner, News360 and NTNU SmartMedia are examples of both commercial and research-oriented news recommender systems.

In order to personalize news articles, item descriptions and user profiles need to be studied and taken into account. The latter one will be discussed in this paper.

The internal functionality of recommender systems is very similar to information retrieval in search engines, in which user profiles are interpreted as queries submitted to an underlying search index. To recommend news articles based on user's interest, the interaction logs are manipulated and stored as user profiles. To build a solid foundation for intelligent recommendations with only user's implicit feedback, different techniques of machine learning are normally applied. After the interaction of user with system, the history of user interactions can serve as training data that is a basis for prediction of new desired news article.

In this paper, we classify supervised and unsupervised machine learning techniques and discuss the features of the user profiles after applying each of them. The challenges of news recommender systems handled by these features are presented. The paper is organized as follows: In section 2 news recommender systems and their details and challenges are described. In section 3 different dimensions of user profiles and machine learning techniques are explained. Features of the user profiles with respect each technique of learning are summarized. In section 4, applying the filtering techniques for content-based, collaborative and different kinds of hybrid system is discussed. The classification of machine learning techniques and their addressed problems are illustrated, before the conclusions are presented in Section 5.

2. News Recommender Systems

News recommender systems share many features with information retrieval systems and human computer interaction as well. Text mining techniques for large scale data sets are needed, and machine learning methods are employed when learning cycles can be built into the systems. In general there are three steps. First of all, data pre-processing such as sampling, dimension reduction, denoising with use of similarity functions are normally applied. Then the text is analyzed through supervised or unsupervised machine learning techniques depending on availability of training data sets. At the end the result is interpreted through for example the F_{1-} measure, ROC or MAE [1].

If we consider news recommender system as search engines, the user profiles can be regarded as long search queries. The system ranks the results on the basis of well the profile matches the descriptions of the news articles. Formally, the appropriateness of recommended news to the user can described by the following utility function [1]:

$$u: C \times S \rightarrow R$$

This function assigns a score *r* for each combination of user *c* and news story *s*. Matrix C indicates the characteristics of the user and S shows the different specifications of available articles such as topic, location, news agency, date and other useful attributes. All different algorithms in recommender systems try to maximize the result matrix. Each entry of R could be any non negative internal between 0 and 1 or 0 and 100 based on the system definition. At the end, an article that maximizes the utility function will be recommended [1]:

$$c' = arg\max_{s \in S} u(c, s)$$

News recommender systems differ in the context of items structures from other recommenders. The structure of news articles is not following any specific format. There are many news articles in a day that have very short life spans while the system must scale to deal with huge volumes of data. Besides, the news recommender system must always recommend interesting articles to the user, though it should not make over-specialize for the target user. [2]

3. User Profiles

The desired user profiles need to have a changing essence and flexible content. These profiles show their preferences towards news articles by modeling the interesting articles. Besides, storing user interactions is a basis to know their favorite topics which last longer and which are only for a short period of time.

This model consists of meta-data such as time and location, which is changing according to the user behavior.

The content of the user profile for this kind of recommender system which has not very structured format is different from others. In order to have an exact and practical model of the user profiles, the system needs to know the behavior of the user including background, interest and goals. These features are changing over time, so considering the temporal parameters such as time and location is crucial [3].

There are three major presentations of terms in the user profile. The first approach is presenting terms as vectors in a vector space model. In order to weigh correctly every single word based on its frequency in every document and in the collection of documents, TF-IDF is often applied. This measure puts more emphasis on one word that appears frequently in one specific document and not in other ones. So it will gain more weight and appointed document, will be retrieved to a target user. But the problem of polysemy (multiple meaning for one word) and synonymy (multiple words for identical meaning) remain. The desired approach reflects cultural and linguistic knowledge of terms and also could use reasoning on their content. As a result, the presentation is more intelligent and is not a simple bag of words and could provide the knowledge about desired terms [1]. The second one is the analysis words in the format of entity. They have meanings and relations, but they suffer from generalization or specialization since there is no hierarchical relationships among the entities [3]. The third one is the semantic analysis that is ontology-based. It has hierarchical relationships between the semantic concepts modeling user interests. The terms that indicate the user interests including their interests that last longer or the ones that appear only for a short time could be enriched by semantic approaches. The advantage of providing ontologies for the user interests is that all the terms or entities are in hierarchical relationships which give more specific detail of user interests at the side of the general ones [3]. The semantic enrichment could benefit from encyclopedic knowledge beside the knowledge of applied documents. So the terms are semantic vectors in word space model [1]. Each of them are indexed by their weights but later will be interpreted semantically by using Wikipedia. It is called Explicit Semantic Analysis (ESA) [4].

The feedback of the user is the other approach of user modeling. In general s/he could communicate and provide their interest towards the news explicitly or implicitly. Explicit feedback is to provide their interest (disaffection) directly to the system. It could be actions such as rating, like or filling the survey through the interface of the application. Implicit feedback includes the interactions such as click on articles (touch in mobile device), scrolling articles using a mouse or a keyboard (swapping in the mobile device), printing or saving articles, copying and posting a part or all of articles, reading articles, forwarding or sharing the articles and providing the qualitative comments on the article. Recommender systems are highly dependent on user feedback. As long as the user interacts with the application, the accuracy of the system may gradually improve. Explicit feedback tends to produce more exact user profiles than what is possible with implicit feedback. Unfortunately, not all users are willing to spend time to provide such feedback, so the implicit signals of the users are normally the basis of the recommendation [5].

Specifying the type of user's interest could help the system to cover all domains of their attention. The long-term interest is more dependent on the user profession and the personal background than what will be traced by the log history. But the short-term interest is the one mostly related to the current trend of public that s/he has communication with. Although depending on the goals, the long-term interest will change gradually. Besides, supervising the context of user's attention could provide good evidence to capture the short-term interest and update their long-term interest time by time. In [6] by defining running context over category and topic, the current user's interest is captured. The old user profile that is the indicator of their long-term interest is updated progressively if there is nothing in common with their current focus. Besides, there should be a balanced focus on the old and new user profile. While keeping the old user profile and over looking the context results in dissatisfaction, giving too much priority to the current context will not cover the news articles that are related to their background and are the basis of their interest. In addition, different time of day (morning, evening) and week (weekdays and weekend) could affect the user profile [7]. Considering the topic of the news articles, target users may have

different desires at different times. As an example, s/he might have more interests in politics and economics in weekdays and focus more on lifestyle news in the weekend [8].

While personalizing the news is desirable, the importance of public trend is not negligible. In [9] based on the frequency of user clicks, public trend could provide the interesting news articles as well. If there are not enough clicks from the user side, then according to their location, public trend of that location is a good indicator to recommend the news. This dimension of the user profile that specifies the location has a key role in recommending news articles. Short-term interests of the user are highly dependent on their location. Location could capture public trend and find similar networks of users as well. Sometimes ignoring the user profile and focus on the context is helpful (in economical news, user profile is not very helpful but the context tracing is more informative), while other times it is better to count only on the user profile (for entertainment section user profile enrichment is much better than context) [10].

As the amount of data explodes, the importance of extracting models and predicting unseen data with machine learning techniques is increasing [11]. There are two major types of learning techniques, supervised and unsupervised. In the former one, an annotated training dataset is provided, whereas in the latter one, the machine explores the data to identify interesting patterns without training data. Below is the list of supervised learning techniques used in recommender systems:

- Decision Trees (C4.5 or KART) handle categorical-nominal and heterogeneous data. It is also able to cope with missing values. Through pre pruning, overfitting will be addressed. It tends to work well with small sized datasets, though the cost of decisions on continuous data streams is high [11, 12].
- Rule-based (RIPPER) can handle multi value features very well. It is decision tree-based and uses rules to categorize new items. It utilizes post pruning to find the best fit for the rule set [13].
- K Nearest Neighbor (KNN) can handle continuous data through Euclidean, Manhattan or Minkowski distance and cope with categorical data through Hamming distance. It is a lazy learner that works well with few instances [14, 15].
- Rocchio and Relevance Feedback: the user profile is regarded as a query [16] and based on the implicit feedback of user, the recommendation will be improved in time.
- Support Vector Machine (SVM): through SVM reduction of sensitivity to the noises and increasing generalization is done. For non linear problem if features are more than instances, linear kernel is good enough to be applied [16, 17].
- Probabilistic methods and Naive Bayes: Bayesian Belief Network with conditional independency is the most applicable one. Multinomial (Bernoulli) and multivariate are two types of Naive Bayes. While in the Bernoulli model absence or presence of a model is checked, in multivariate one the number of occurrences of a term will be calculated [18, 19].
- Neural Network: Single layer perceptron and multi layer for non linear separable problems are the samples of applied neural network in the recommender systems [20].

Below is the list of unsupervised learning techniques:

- Probabilistic methods: If the structure of Bayesian network is not known then building the DAG Bayesian with scoring function, constraint based learning or Conditional Independency can be applied. The last one has more efficiency [21]. The other techniques such as Bayesian Hidden Score (pairwise learning) and graph-based learning have been applied in [22].
- Neural Network: Self Organizing Map (Kohonen) and Restricted Boltzmann Machine belong to the category of unsupervised learning [20].
- Clustering: flat clustering by k-means algorithm deals with the categorical data and the most frequent term will be the centroid. In the hierarchical clustering, the other type of clustering, divisive is more accurate than agglomerative. There are two approaches to label clusters. The first one is differential that through feature selection a label with a higher score will be chosen. The second one is inter clustering that the closest one to the title or the higher weight to the centroid of the cluster will be chosen as the label. The drawback of cluster-internal labeling is disability to distinguish between words which are frequent in the whole clusters and the ones that are frequent only in one specific cluster. Labeling in hierarchical clustering due to the dependent definitions of parent, child and sibling is more complicated [16].

Table 1 shows the applied machine learning techniques to build up a user profile.

4. Applying User Profiles in Recommender Systems

There are different approaches to filter out the information. Content-based and collaborative filtering are the most applicable ones. In content-based filtering, the concept of news articles will be analyzed. Then according to the content of the user profile (i.e. characteristic of read articles), similar articles are predicted and presented to the user. In the content-based filtering, the utility function is:

$$u(c,s) = \text{score}(\text{contentbased user profile}(c), \text{item content}(s))$$

If each of the content of the user profile and item profile is represented by TF-IDF weight, then the scoring function could be calculated through cosine similarity of vectors of the weight. To achieve the accurate prediction, attributes of news articles that have been counted on, are important. Since the nature of news article is unstructured, extracting relevant and important features has a key role in content-based filtering. If the articles are categorized with minimum misclassification error, then storing interesting news articles in the user profile is much easier and consequently, recommendations are of higher quality. Bayesian Networks can be utilized well for learning user profiles based on the articles that have been read. It can model profiles of the users through ignoring missing data and considering conditional dependency in one specific category of news articles. It can provide probabilities of each attribute of article by its nodes. The modeled domain includes continuous data. Then similarity of the user profile based on predicted attributes of article and available news articles is computed and the ones with the highest score will be recommended. If another technique such as Naive Bayes (Bernoulli Model) is applied for modeling user behavior, the output is binary as it is considering absence or presence of terms

regardless of their conditional independency [1]. It can suggest the new item to the target user by comparing the new item's characteristics to the terms in the user's profile. But if there is not enough attributes, content-based filtering is normally not the most efficient one. If the user is new to the system it cannot recommend anything as there is no content of their profile available. Besides, it causes lack of serendipity due to providing too many similar news articles to the user. Considering the collaborative approach for filtering information, there are two different models, memory-based and model-based. Memory-based utilizes the log

Table 1. ML techniques and features of user profiles

ML Techniques	User Profile Features
Decision Tree (C4.5)	Semantic enrichment can be handled at entity level, but in the beginning of building the user profile or for capturing short-term interest [13, 23].
Rule-based (RIPPER)	Semantic enrichment can be handled at entity level. More interesting categories of news may be predicated through rules [1].
KNN	Captures the short term interest of user and popularity of the item among a group of user.
Rocchio and Relevance Feedback	User profiles are regarded as queries, the system improves over time from relevance feedback of the user [16].
Support Vector Machine	It outperforms KNN,C4.5 and Rocchio [16] with the Reuters dataset
Probabilistic methods and Naive Bayes	Bernoulli works well with small sizes of data set and multinomial works well in large sizes of datasets. DAG captures the dependency of items in more detailed capturing interest, vigorous towards missing data and could disregard noisy data. BHS and graph-based capture online interest of the user [22]
Neural Network	It can represent details of the user's interest through deep learning of three layer perceptron [24].
Clustering	The content of the items are clustered and then item-based collaborative is implemented on the output. Fuzzy membership over the k-means. Similarity of the item-rating matrix, the group-rating matrix (MovieLense) Hierarchical clustering for the news groups (LDA for small dataset and PLSI for large dataset) [25]

history of all users and put top-N similar users who have the same taste about the news articles into one specific group.

Then to provide the latest and interesting news articles to the target user, it filters out users with the same interest and recommends the new articles that have been read by them. It is working with a matrix of user's profile and all the news articles. It is possible to apply K Nearest Neighbor (through neighborhood measurement) to find the closest users to the current active user. The other approach is applying similarity measurement like cosine similarity or Pearson correlation, which provide the new item for the target user if it has similarity with previous chosen items. It can help us find similar users or items regarding to the context of memory [23].

The other type of collaborative filtering is model-based. It is more scalable and much faster than memory based collaborative filtering. Through this type of filtering not all the dataset will be traced and investigated, but only some information will be modeled. As finding the similarity between users or news articles (users with the same interest in the specific news or two similar news articles that are interesting to one specific user) is not feasible due to lack of labeled data in the training phase, clustering of news or users could be a practical solution. With the Google News dataset, clustering is done on the basis of users' clicks on different news article. Through clustering, latent factors (latent semantic analysis) can be revealed. Consequently, ignoring the hidden values will result in a very poor accuracy. It could be helpful to distinguish hidden variables through the clustering and provide more accurate prediction of news articles [23]. One the technique to implement this approach is building up the matrix of users and item as matrix factorization. The matrix of users and news articles is suffering from sparsity, since there are several positions that users do not provide any feedback. To find the hidden variables that affect the recommendation as well, *UV* decomposition (it is one instance of Singular Value Decomposition) is possible to be applied. If the utility matrix M is $n \times m$ (n indicates the user and m indicates the news articles), then *UV* decomposes it multiplication of two different matrixes including $n \times d$ and $d \times m$:

$$M \rightarrow U \times V$$

RMSE is a common tool to measure the accuracy of prediction blank entries in M considering the product UV.

Although it is working much faster than memory based, it is less exact than it. In spite of all the applicable different approaches of collaborative filtering, it cannot make the accurate prediction for the new user or the new item (cold-start problem). The core of all the algorithms is dependent on the group of users (or items) in order to find the proper match for the target user. Consequently it has nothing to present to the user with unique taste.

As each of these filtering techniques has its own problems and challenges in recommender systems, a hybrid system is often preferred. It takes into account both filtering in predefined step and could overcome drawback of each. Considering two techniques of filtering (content-based and collaborative), the order of combination of them might be important to build a hybrid system. Although in some techniques of hybridization, the order is not a matter. The techniques that order is not important are [26]:

Mixed: the result from both techniques will be presented in one grouped or separate list. It has been utilized in [27] to provide the TV shows to the users. The mixed hybrid system provides recommendations based on the characteristics of each show and preferences of other users.

Weighted: The score for each technique is computed, and the weighting of final score will be the basis for the recommendation. In personalized Tango (P-tango) for online newspapers, equal weights are assigned to both filtering techniques. Gradually each weight is increasing regarding the user rating. Based on the rating, the absolute error is computed and is decreasing through the better recommendation.

Switching: This technique uses some criterion to switch between filtering techniques and based on the specific chosen filter, recommends the item. In the DailyLearner switching hybrid system, content-based filtering with k nearest neighbor is first applied. If it does not produce sufficient recommendations, collaborative filtering takes advantage of similar users' interests to recommend desired items. In another system, item-based collaborative filtering is triggered if the accuracy of the content-based filtering part is low [28].

Feature combination: The technique takes advantage of one filtering type such as collaborative filtering as feature allied with data. Then content-based filtering is applied. Through this kind of hybrid system, the absolute dependency on users is dropped by applying collaborative filtering as a feature combination. In the movie recommender domain [29], the RIPPER algorithm is implemented with item features and users rating.

There are three other models of hybrid systems that are ordered by their intrinsic structure:

Feature augmentation: One of the filtering techniques is applied to compute rating scores or to classify items. The output of this filtering is the input for the other filtering technique. In Libra system, content-based filtering through Naïve Bayes is done on data that comes from Amazon. The data from Amazon that show related authors and titles were implemented using collaborative filtering. Collaborative filtering is done first.

Meta-level: It provides a model through one of the filtering methods as an input for the other one. The model is the complete one, not a learned model like feature augmented techniques. In Fab [30] at first by means of relevance feedback and the Rocchio algorithm, collections of items (the need of users in mass of dataset in web) are composed (content-based). K-nearest neighbor is then used with collaborative filtering to complete the recommendations. Meta-level is the only ordered technique that applies content-based filtering first.

Cascade: Approximately similar to the other ordered techniques, it refines the result of candidates that have been filtered by the previous technique. But if the items in the first filtering have very low priorities, they will not be in the second filtering stage. In fact, the second filtering step is only applied to provide more accurate recommendations and if an item has not enough rating score, it will not be in the second phase. Fab [31] is the example of this technique. With collaborative filtering on the selection stage, the items are chosen with an exact score and presented to the user.

According to the implemented hybrid systems in news recommender system (such as Daily Learner), switching schema is the most common strategy. It can start with content-based filtering and utilize Naive Bayes to categorize the news articles based on the content of the articles and apply item-based collaborative filtering to calculate the similarity between the news articles and the user profile. On the other hand, it is also possible to apply collaborative filtering to find the closest users to the active user (through KNN) and then with content-based filtering identify much more similar items based on the similarity computation of user profile and news articles.

Table 2 shows the applied machine learning techniques to deal with the issues of news recommender systems.

Table 2. Machine learning techniques and challenges addressed

ML Techniques	Challenges addresses of news recommender system
Decision Tree (C4.5)	Capturing short term interest [1].
Rule-based (RIPPER)	Serendipity can be supported with new category reasoning [32].
KNN	Short-term interests and provide the latest news to the user based on their interests [1].
Rocchio and Relevance Feedback	Handling long-term interest of the user [1].
Support Vector Machine	Sparse Problem and huge data after a long time usage of the application[33].
Probabilistic methods and Naive Bayes	Handling long-term interest of the user Sparse problem Noisy data Cold Start

	Precious interest of the user [28].
Neural Network	Short term and long term [34]. Tied Boltzmann with residual parameter could outperform on non cold-start problem in comparison with simple method of collaborative filtering, Pearson correlation for the items. It also is competitive with the cold-start problem in content-based filtering. (Netflix) Changing interest of the user [24].
Clustering	Cold start Through fuzzy membership new and interesting news articles are possible to be represented to the user [25].

5. Conclusion

The news recommender system is somewhat different from other recommender systems. It is used to provide a variety of personalized news articles that have very short life spans. In addition the range of the user's interests is wide and changing over time and contexts. These characteristics necessitate very dynamic analyses of user profiles.

In this paper the distinguishable characteristics that affect recommendation strategies are assessed. The user feedback on recommended items is one of them. Different algorithms of machine learning (that fall into the categories of supervised and unsupervised) are discussed to build up user profiles. On the other hand, as the user profile is dependent on the whole framework of filtering methods, the techniques are also studied. They utilize user profiles in diverse ways which affect the accuracy of the corresponding recommendations.

References

[1] Ricci, F., et al., Recommender Systems Handbook. 2010: Springer-Verlag New York, Inc. 842.

[2] Özgöbek, Ö., J. A. Gulla,, R. C. Erdur, A Survey on Challenges and Methods in News Recommendation, in In Proceedings of the 10th International Conference on Web Information System and Technologies April 2014: Barcelona.

[3] Bouneffouf, D., Towards User Profile Modelling in Recommender System. 2013.

[4] Gabrilovich, E. and S. Markovitch, Computing semantic relatedness using Wikipedia-based explicit semantic analysis, in Proceedings of the 20th international joint conference on Artifical intelligence. 2007, Morgan Kaufmann Publishers Inc.: Hyderabad, India. p. 1606-1611.

[5] Jon Atle Gulla, J.E.I., Arne Dag Fidjestøl, John Eirik Nilsen, Kent Robin Haugen, and Xioameng Su, Learning User Profiles in Mobile News Recommendation. Journal of Print and Media Technology Research, September 2013. Vol II, No. 3: p. pp. 183-194.

[6] Jon Atle Gulla, A.D.F., Xiaomeng Su and Humberto Castejon, Implicit User Profiling in News Recommender Systems, in In Proceedings of the 10th International Conference on Web Information System and Technologies April 2014: Barcelona.

[7] Abel, F., et al., Analyzing user modeling on twitter for personalized news recommendations, in Proceedings of the 19th international conference on User modeling, adaption, and personalization. 2011, Springer-Verlag: Girona, Spain. p. 1-12.

[8] Adomavicius, G. and A. Tuzhilin, Context-aware recommender systems, in Proceedings of the 2008 ACM conference on Recommender systems. 2008, ACM: Lausanne, Switzerland. p. 335-336.

[9] Liu, J., et al., Personalized news recommendation based on click behavior, in Proceedings of the 15th international conference on Intelligent user interfaces. 2010, ACM: Hong Kong, China. p. 31-40.

[10] Bellogín, A., et al. Discovering Relevant Preferences in a Personalised Recommender System using Machine Learning Techniques. in Preference Learning Workshop (PL 2008), at the 8th European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2008). 2008.

[11] Witten, I.H., E. Frank, and M.A. Hall, Data Mining: Practical Machine Learning Tools and Techniques. 2011: Morgan Kaufmann Publishers Inc. 664.

[12] VENKATADRI.M, L.C.R. A Comparative Study On Decision Tree Classification Algorithms In Data Mining. 2010; Available from: https://www.academia.edu/1374211/A_Comparative_Study_On_Decision_Tree_Classification_Algorithms_In_Data_Mining.

[13] Pazzani, M.J. and D. Billsus, Content-based recommendation systems, in The adaptive web, B. Peter, K. Alfred, and N. Wolfgang, Editors. 2007, Springer-Verlag. p. 325-341.

[14] Deokar, S., WEIGHTED K NEAREST NEIGHBOR. 2009.

[15] Webb, G., M. Pazzani, and D. Billsus, Machine Learning for User Modeling. User Modeling and User-Adapted Interaction, 2001. 11(1-2): p. 19-29.

[16] Manning, C.D., et al., Introduction to Information Retrieval. 2008: Cambridge University Press. 496.

[17] Prügel-Bennett, M.A.G.a.A., Building Switching Hybrid Recommender System Using Machine Learning Classifiers and Collaborative Filtering. IAENG International Journal of Computer Science.

[18] Margaritis, D., Learning Bayesian Network Model Structure. 2003, University of Pittsburgh.

[19] Barber, D., Bayesian Reasoning and Machine Learning. 2010.

[20] Peretto, P., An Introduction to the Modeling of Neural Networks. 1992: Cambridge University Press.

[21] Kotsiantis, S.B., Supervised Machine Learning: A Review of Classification Techniques, in Proceedings of the 2007

conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies. 2007, IOS Press. p. 3-24.

[22] Bian, J., et al., Exploiting User Preference for Online Learning in Web Content Optimization Systems. ACM Trans. Intell. Syst. Technol., 2014. 5(2): p. 1-23.

[23] Rajaraman, A. and J.D. Ullman, Mining of Massive Datasets. 2011: Cambridge University Press. 326.

[24] Gunawardana, A. and C. Meek, A unified approach to building hybrid recommender systems, in Proceedings of the third ACM conference on Recommender systems. 2009, ACM: New York, New York, USA. p. 117-124.

[25] Mouton, C., Unsupervised Word Sense Induction from Multiple Semantic Spaces with Locality Sensitive Hashing, in International Conference RANLP. 2009.

[26] Burke, R., Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction, 2002. 12(4): p. 331-370.

[27] Cotter, P. and B. Smyth, PTV: Intelligent Personalised TV Guides, in Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence. 2000, AAAI Press. p. 957-964.

[28] Ghazanfar, M.A. and A. Prugel-Bennett, An Improved Switching Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering. International Multiconference of Engineers and Computer Scientists (Imecs 2010), Vols I-Iii, 2010: p. 493-502.

[29] Basu, C., H. Hirsh, and W. Cohen, Recommendation as classification: using social and content-based information in recommendation, in Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence. 1998, American Association for Artificial Intelligence: Madison, Wisconsin, USA. p. 714-720.

[30] Balabanovi, M., #263, and Y. Shoham, Fab: content-based, collaborative recommendation. Commun. ACM, 1997. 40(3): p. 66-72.

[31] Balabanovi, M. and #263, An adaptive Web page recommendation service, in Proceedings of the first international conference on Autonomous agents. 1997, ACM: Marina del Rey, California, USA. p. 378-385.

[32] Markward Britsch, N.G., Michael Schmelling, Application of the rule-growing algorithm RIPPER to particle physics analysis. 2008.

[33] Anatole Gershman, T.W., Eugene Fink, Jaime Carbonell, News Personalization using Support Vector Machines.

[34] Kyo-Joong Oh, W.-J.L., Chae-Gyun Lim, Ho-Jin Choi, Personalized News Recommendation using Classified Keywords to Capture User Preference, in Advanced Communication Technology (ICACT). 2014.

News2Images: Automatically Summarizing News Articles into Image-Based Contents via Deep Learning

Jung-Woo Ha
NAVER LABS
NAVER Corp.
Seongnam, 463-867, Korea
jungwoo.ha@navercorp.com

Dongyeop Kang
NAVER LABS
NAVER Corp.
Seongnam, 463-867, Korea
dongyeop.kang@navercorp.com

Hyuna Pyo
NAVER LABS
NAVER Corp.
Seongnam, 463-867, Korea
hyuna.pyo@navercorp.com

Jeonghee Kim
NAVER LABS
NAVER Corp.
Seongnam, 463-867, Korea
jeonghee.kim@navercorp.com

ABSTRACT

Compact representation is a key issue for effective information delivery to users in mobile content-providing services. In particular, it is more severe when providing text documents such as news articles on the mobile service. Here we propose a method for generating compact image-based contents from news documents (News2Image). The proposed method consists of three modules for summarizing news into a few key sentences based on the sematic similarity and diversity, converting the sentences into images, and generating contents consisting of sentence-embedded images. We use word embedding for document summarization and convolutional neural networks (CNNs) for sentence-to-image transformation. These image-based contents improve the readability, thus effectively delivering the core contents of the news to users. We demonstrate the news-to-image content generation on more-than one million Korean news articles using the proposed News2Image. Experimental results show our method generates better image-contents semantically related to the given news articles compared to a baseline method. Furthermore, we discuss some directions for applying News2Images to a news recommendation system.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models and Selection process

General Terms

Algorithms, Experimentation

Keywords

news-to-images, summarization, image-based contents, automatic content generation, deep learning, mobile service.

1. INTRODUCTION

Mobile devices have become one of the most important channels for information delivery replacing personal computers (PCs). People search news articles, buy items, and see videos using their smartphones and tablets without respect to time and place. Although the mobile devices have various advantages in portability and convenience compared to PCs, however, the restriction on their display size requires more compact representation or visualization of information while minimizing loss of the information delivered. In particular, it is more critical to provide text documents composed of many sentences such as news articles on mobile services. For overcoming this limitation, image-oriented tiny blog services such as Pikicast[1] are operated through mobile applications and mobile web sites with popularity. These image-based contents can not only deliver the core contents in a short period of time but also arouse users' interests compared to text documents. However, these contents are manually generated by human experts or writers.

Deep learning is a machine learning method based on neural networks with a deep architecture [6]. Deep learning approaches have showed amazingly successful reports in diverse domains including speech recognition [2], image and video classification [5], natural language processing [12], and recommendation [11, 13] for recent several years, thus being considered as a most promising framework for big data analysis. The main advantage of deep learning against other machine learning methods is that features are automatically constructed by the learning [6]. Deep learning models can automatically construct features, represented with real-valued vectors regardless of the characteristics of input data, for the learning process. These constructed features can be used for other learning tasks as input data. In particular, it is reported that the features from word embedding networks [10] and deep convolutional neural networks [5] outperform manually hand-crafted those in many studies. Furthermore, this vector representation is suitable for characterizing a common semantic space for learning from multimodal data [14].

Here we propose a method for automatically generating image-based contents from news articles using deep learning (News2Images). Generating image-based contents from news documents includes three subtasks: i) summarizing news documents into key sentences, ii) retrieving images corresponding to the contents of the summarized sentences, and iii) generating image-based contents for enhancing users' convenience and interests. Therefore, the proposed News2Images method consists of three modules dedicated to each subtask. The summarization module extracts multiple core sentences from a given news article using a single document summarization method. For extracting

RecSys 2015, September 16–20, 2015, Vienna, Austria.

[1] www.pikicast.com

Figure 1. An example of the image-based contents generated from a news document by News2Images. Left box includes an original online news document and right box represents the contents summarizing the news into three images. Red sentences in the left box are key sentences extracted by summarization and they are located in the black rectangle below the retrieved images in the right box.

key sentences, we define a score considering both the similarity to the core news contents and the diversity for the coverage on the entire contents of the news. The similarity and the diversity are computed using sentence embedding based on word2vec [10]. The image retrieval module searches the images semantically associated with the sentences extracted by the summarization module. The semantic association between a sentence and an image is defined as the cosine similarity between the sentence and the title of the news article which the image is attached in. Also, we use the hidden node values of the top fully connected layer of the convolutional neural networks (CNNs) [4] for each image as an image feature. Finally, the image-based content module generates a set of new images by synthesizing a retrieved image and the sentence corresponding to the image. These image-based contents generated can improve the readability and enhance the interests of mobile device users, compared to text-based news articles. The proposed News2Images has the originality in aspect of generating new contents suitable for mobile services by summarizing a long news document into not sentences but images even if there exist many methods for summarization [9] or text-to-image retrieval [1]. Figure 1 presents an example of the image-based content consisting of three synthesized images generated from a Korean online news article.

We evaluate the proposed News2Images on a big media data including more-than one million news articles served through a Korean media portal website, NAVER[2], in 2014. Experimental results show our method outperforms a baseline method based on word occurrence in terms of both quantitative and qualitative criteria. Moreover, we discuss some future directions for applying News2Images to personalized news recommender systems.

2. DEEP LEARNING-BASED FEATURE REPRESENTATION

Most news articles consist of a title, a document, and attached images. Mathematically, a news article x is defined as a triple $x = \{\mathbf{t}, S, V\}$, where $\mathbf{t}$, S, and V denote a title, the set of document sentences, and an image set. V can be an empty set. A title $\mathbf{t}$ and a document sentence $\mathbf{s}$, $\mathbf{s} \in S$, are represented as a vector of word features such as occurrence frequency or word embedding. An image $\mathbf{v}$, $\mathbf{v} \in V$ is also defined as a vector of visual features such as Scale invariant feature transform (SIFT) [8] or CNN features. For representing a news article with a feature vector, we use deep learning in this study.

Many recent studies have reported that the hidden node values generated from deep learning models such as word embedding networks and CNNs are very useful for diverse problems including image classification [5], image descriptive sentence generation [14], and language models [12].

Formally, a word $\mathbf{w}$ is represented as a real-valued vector, $\mathbf{w} \in \Re^d$, where d is the dimension of a word vector. The vector value of each word is learned from a large corpus by word2vec [10]. This distributed word representation, called word embedding, is to not only characterize the semantic and the syntactic information but also overcome the data sparsity problem [6, 10]. It means that two words with similar meaning are located at a close position in the vector space. A sentence or a document can be represented as a real-valued vector as well. Sentence or document vectors can be generated by learning of deep networks, or they are calculated by pooling the word vectors included in the sentences. Here a sentence vector is calculated by average pooling:

$$s_i = \frac{1}{|s|} \sum_{w \in s} w_i , \quad (1)$$

where w and s denote a word and the set of words included in a sentence. Also, s_i and w_i are the i-th element of embedding vector $\mathbf{s}$ and $\mathbf{w}$ corresponding to s and w, respectively. Simple average pooling leads to lose sequence information of words. Therefore, the concatenation of multiple word vectors and the sliding window strategy can be used instead of simple pooling.

Image features can be generated for an input image by the CNNs learned from a large-scale image database. Typically, the hidden node values of the fully connected layer below the top softmax layer of CNNs are used as features. The CNN image features are also represented as a (non-negative) real-valued vector and they are known to be distinguishable for object recognition.

[2] www.naver.com

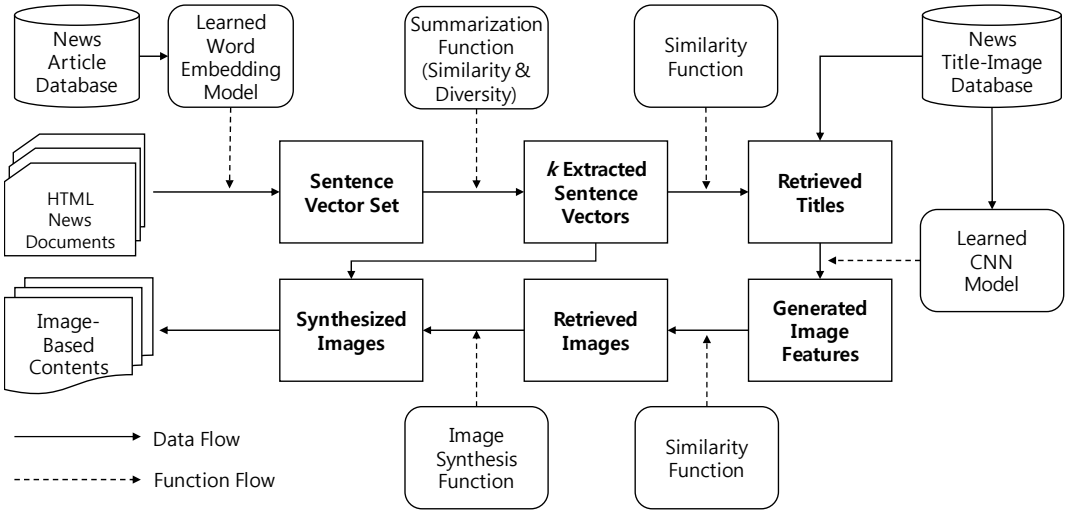


Figure 2. Overall flow of generating image-based contents from a news article via News2Images

3. NEWS-TO-IMAGES

News2Images is a method of generating image-based contents from a given news document using summarization and text-to-image retrieval. News2Images consists of three parts including key sentence extraction based on the single document summarization, key sentence-related image retrieval by associating images with sentences, and image-based content generation by synthesizing sentences and images. Figure 2 shows the overall framework of News2Images.

3.1 News Document Summarization

Document summarization is a task of automatically generating a minority of key sentences from an original document, minimizing loss of the content information [9]. Two approaches are mainly used for document summarization. One is abstraction which is to generate a few new sentences. Abstraction more precisely summarizes a document but still remains a challenging issue. The other is extraction, to select some core sentences from a document, and we use the extraction approach in this study. Also, the news summarization in this study belongs to single document summarization [7]. We assume two conditions for the summarization:

i) A news title is the best sentence consistently representing the entire content of the news.

ii) A news article consists of at least two sentences and the entire content is built up by composing its sentences' content.

For precisely summarizing a news document, thus, it is required that a summarized sentence set consists of the sentences not only semantically similar to its title but also covering the entire content with diverse words. We call the former similarity and the latter diversity.

Formally, a document S is defined as a set of its sentences, $S = \{\mathbf{s}_1, ..., \mathbf{s}_M\}$, where M denotes the number of the sentences included in S. The i-th sentence $\mathbf{s}_i$ is represented as a real-valued vector, $\mathbf{s}_i \in \Re^d$, where d is the vector size, by word2vec and average pooling. Then, document summarization is formulated with

$$S_k^* = \arg\max_{S_k \subset S} \{\alpha \cdot f(S_k, S) + (1-\alpha) \cdot g(S_k, S)\} = \arg\max_{S_k \subset S} \{\alpha \cdot f(S_k, t) + (1-\alpha) \cdot g(S_k, S)\}, \quad (2)$$

$$\text{s.t. } f(S_k, S) = \sum_{\mathbf{s} \in S_k} f(\mathbf{s}, S) \text{ and } g(S_k, S) = \sum_{\mathbf{s} \in S_k} g(\mathbf{s}, S),$$

where $\mathbf{t}$ denotes the title of S, S_k and S_k^* are the set of k sentences extracted and an optimal set among S_k. $f(S_k, S)$ and $g(S_k, S)$ denote the similarity and the diversity functions, and α is the constant for moderating the ratio of two criteria.

The similarity $f(\mathbf{s}, \mathbf{t})$ between a given sentence $\mathbf{s}$ and a news title $\mathbf{t}$ is defined as the cosine similarity between two sentence embedding vectors:

$$f(\mathbf{s}, \mathbf{t}) = \frac{\mathbf{s} \cdot \mathbf{t}}{\|\mathbf{s}\| \|\mathbf{t}\|}. \quad (3)$$

For calculating the diversity, we partition the sentences of S into multiple subsets using a clustering method. Because a sentence vector implicitly reflects syntactic and semantic information, multiple semantically distinctive subsets are generated by clustering. For the j-th cluster C^j, we calculate the cosine similarity between all the sentences in C^j and the centroid of C^j. Because the cosine similarity can be negative, we consider a negative value as zero. This value is defined as the diversity:

$$g(\mathbf{s}, C^j) = \frac{\mathbf{s} \cdot \mathbf{c}^j}{\|\mathbf{s}\| \|\mathbf{c}^j\|}, \quad (4)$$

where $\mathbf{c}^j$ denotes the centroid vector of C^j.

Finally, k sentences with the largest value defined in (2) are extracted as the summarization set for the given document. Here we set k to three, which means that a news article is summarized into three image-based contents.

3.2 Sentence-to-Image Retrieval

The second subtask is to retrieve the images representing semantics similar to the extracted sentences. Because we use the images attached in news articles, the title of a news including an image can be used as a description sentence of the image.

Therefore, the semantic similarity of an image to an extracted sentence is calculated by measuring the similarity between the image title vector and the sentence vector.

Formally, when an image feature vector set, $V=\{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$, is given, the images similar to an extracted sentence $\hat{\mathbf{s}}$ are extracted:

$$\mathbf{v}^* = \arg\max_{\mathbf{v}\in V} \left\{ f\left(\hat{\mathbf{s}}, \mathbf{t}(\mathbf{v})\right) \right\} = \arg\max_{\mathbf{v}\in V} \left\{ \frac{\hat{\mathbf{s}} \cdot \mathbf{t}(\mathbf{v})}{|\hat{\mathbf{s}}||\mathbf{t}(\mathbf{v})|} \right\}, \quad (5)$$

where $\mathbf{t}(\mathbf{v})$ denotes the title of an image $\mathbf{v}$.

Due to the diversity, sentences which are not directly related to the title may be extracted as a core sentence. We assume that a title is "*Yuna Kim decided to participate in 2013 world figure skating championship*", and two extracted sentences are "*Yuna Kim will take part in the coming world figure skating championship*" and "*The competition will be held in February*." In this case, the title is not semantically similar to the second sentence. Thus it is difficult to associate the second sentence with Yuna Kim's images. For overcoming this, we can additionally use the title vector of the news articles given as a query for pooling word vectors into a sentence vector. The use of the news title does not influence the summarization because the title vector is reflected on all the sentence vectors.

Instead of $\mathbf{v}^*$, we can generate a new image vector $\hat{\mathbf{v}}$ by averaging the vectors of top K images with the large similarity value. Then, $\mathbf{v}^*$ is selected as follows:

$$\mathbf{v}^* = \arg\max_{\mathbf{v}\in V} \left\{ f\left(\hat{\mathbf{v}}, \mathbf{v}\right) \right\}, \quad (6)$$

$$\hat{v}_i = \frac{R(\mathbf{v})}{\sum_{\mathbf{v}\in V_K} R(\mathbf{v})} v_i, \quad (7)$$

where v_i is the i-th element of $\mathbf{v}$ and $R(\mathbf{v})$ denotes a weight function proportional to the similarity rank. An image more similar to $\hat{\mathbf{v}}$ has a larger $R(\mathbf{v})$.

3.3 Image-Based Content Generation

Readability is a main issue of mobile content service. Therefore we generate new image-based contents instead of using the retrieved images for improving the readability and enhancing the users' interests. An image-based content includes continuous series of synthesized images where the retrieved images and their corresponding sentences are merged. Figure 1 illustrates an example of the image-based contents from a news document.

4. EXPERIMENTAL RESULTS

4.1 Data and Parameter Setting

We evaluate the proposed News2Images on a big media data including over one million Korean news articles, which are provided by a media portal site, NAVER, in 2014. In detail, the word vectors are learned from all the news documents and the CNN models for constructing image features are trained from approximately 220 thousands of news images, which are related to 100 famous entertainers, movie stars, and sports stars. Also, 6,967 news articles are used as the validation set for evaluating the performance. Three key sentences were extracted from a news article including more than three sentences and we used all the sentences in the news consisting of less than three sentences. Then, 20,224 image-based contents were generated from validation news data in total.

We used the word2vec for word embedding and modified GoogleNet implemented in Caffe for CNN features [4]. The word vector and image feature sizes are 100 and 1024, respectively. For error correction in learning CNNs, we set the label of an image to the person name in the image. Thus, the size of the class label set is 100. The learned CNN model for generating image features yields 0.56 and 0.79 as Top-1 and Top-5 classification accuracies, respectively. This indicates that the generated image features are distinguishable enough to be used for associating images and sentences. The number of clusters for the diversity in summarization was set to 3 and the constant moderating the similarity and the diversity is 0.9.

For comparisons, we used a word occurrence vector based on TF/IDF as a baseline in computing the similarity between sentences and titles, instead of a word embedding vector. TF/IDF has been widely used for text mining, and thus we can verify the effects of deep learning-based word features.

Table 1. Accuracy of the baseline method and News2Images

Classification	Baseline (TF/IDF)	News2Images
Correct #	14,020/20,224	**18,908/20,224**
Accuracy	0.693	**0.935**
Cosine Similarity	0.636	**0.866**

We set the number of images for averaging in (6), M to 1 both two methods. The window size of the words is 1. Both methods use news titles in pooling word vectors into sentence vectors.

4.2 Content Generation Accuracy

Human efforts are still essential for precisely measuring how similar the generated image-based contents are semantically to the

Table 2. Accuracies according to the usage of news titles

News title	No used	Used
Correct #	13,896/20,224	**18,908/20,224**
Accuracy	0.687	**0.935**

Table 3. Accuracies according to the size of retrieved images size for generating a new image feature

Image size	K=1	K=3
Correct #	**18,908/20,224**	18,791/20,224
Accuracy	**0.935**	0.929

Table 4. Accuracies according to the weight for proper nouns

Proper noun weight	PW = 1.0	PW=10.0
Correct #	18,908/20,224	**19,191/20,224**
Accuracy	0.935	**0.950**

PW denotes the weight of proper nouns.

Table 5. Accuracies according to word vector window sizes

Window size	$\|W\|$=1	$\|W\|$=3
Correct #	**18,908/20,224**	18,743/20,224
Accuracy	**0.935**	0.927
Cosine Similarity	**0.866**	0.833

$|W|$ denotes the number of concatenated word vectors.

news document given as a query. Instead of manual evaluation by humans, we consider a classification problem as the similarity evaluation. That is, for a given extracted news sentence, we consider that the retrieved image is similar to the sentence when the persons referred in the sentence exist in the image. It is reasonable because this means the method provides diverse images of a movie star for users when a user reads a news about the star.

Table 1 compares the classification accuracy of the baseline and the proposed method. As shown in Table 1, News2Images outperforms the baseline method. This indicates the word embedding features used in News2Images more precisely represent semantics, compared to TF/IDF-based features. Also, we compared the cosine similarity between the titles of the retrieved images and the extracted sentences using their word embedding vectors. The values are averaged on the titles of 20,224 retrieved images. We can find that our method retrieves the images more semantically similar to the extracted sentences.

4.3 Effects of Parameters on Performance

We compare the accuracies of the generated contents under four parameters including i) the use of news title for pooling word vectors into a sentence vector, ii) the number of retrieved images for an image feature, iii) the weight for proper nouns, and iv) the size of concatenated word vectors. Table 2 presents the accuracy improvement when the title of the summarized news documents is used. We found that the use of the news title dramatically improves the accuracy as 30% compared to the case in which the titles are not used. Interestingly, News2Images not using titles provides the similar performance to the baseline method using titles. Table 3 shows the effects of averaging multiple image features on sentence-to-image retrieval. This indicates that generating a new image feature from multiple image features has no effect on enhancing the performance. To give more weight to proper nouns can improve the quality of the image-based content generation because proper nouns are likely to be a key content of the news. The results in Table 4 support this hypothesis. The number of concatenated word vectors rarely influences the accuracy. We indicate that the information on word sequences is not essential to classify the person in the images from Table 5.

4.4 Image-Based Contents as News Summarization

Figure 3 illustrates good and bad examples of image-based

Sentences	News2Images	Baseline
Park, the home run leader of KBO, hit the 34th home run in this season.		
Son of Leverkusen played as a starter forward in this game for 60 minutes until substituted with Yurchenko		
Today, Ryu pitched 7 innings, allowed two runs and 9 hits, and got 7 kills against the Chicago Cubs at the home game, and thus ERA becomes 3.39.		
Lee, Hyori is practicing yoga with a grave look in the released photo.		
Chu, Soohyun showed her bodyline at the swimming pool scene in the 18th episode of the drama.		

Figure 3. Examples of image-based contents generated from the summarization sentences extracted from news articles by News2Images and the baseline method. Images with a red border are very similar to the sentences. Blue bordered images include the persons referred in the given sentences but represent contents different from the sentences.

contents from news articles. Most of the images are related to the news contents but the sentences including polysemy or too many words are occasionally linked to images not relevant to the sentences. This is caused that one word is represented as only one vector regardless of its meaning. Also, the representation power of pooling-based sentence embedding can be weaken due to the property of average pooling when a sentence consists of too many words.

5. DISCUSSION

We proposed a new method for summarizing news articles into image-based contents, News2Images. These image-based contents are useful for providing the news for mobile device users while enhancing the readability and interests. Deep learning-based text and image features used in the proposed method improved the performance as approximately 24% of the classification accuracy and 0.23 of the cosine similarity compared to the TF/IDF baseline method. Our study has an originality in aspect of generating new image contents from news documents even if many studies on summarization or text-to-image retrieval have been reported.

This method can be applied to a personalized news recommender system adding user preference information such as subject categories and persons preferred by a user and feedback information into the method. In detail, we can give a weight to words related to subjects or persons preferred by a user when generating sentence vectors. This strategy allows the sentences which the user is likely to feel an interest in to have higher score in summarization and retrieval, thus exposing the photos which the user prefers.

Evaluation should be also improved. Although we evaluate the proposed method with the cosine similarity-based measure and the classification accuracy, it has a limitation for precisely measuring the similarity between the news articles and the image contents generated. It is required to make a ground truth dataset by humans, which not only helps to more precisely evaluate the model performance and can be used as a good dataset for recommendation as well as image-text multimodal learning. Furthermore, we will verify the effects of News2Images on the improvements of the readability through human experiments as future work.

The proposed method can be improved by adding the module of efficiently learning a common semantic hypothesis represented with sentences and images using a unified model [14].

ACKNOWLEDGMENTS

6. REFERENCES

[1] Datta, R., Joshi, D., Li, J. and Wang, J. Z. 2008. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys* (*CSUR*). 40, 2. 5.

[2] Hinton, G. et al. 2012. Deep neural networks for acoustic modeling in speech recognition, *IEEE Signal Processing Magazine*. 29, 6. 82-97.

[3] Irsoy, O. and Cardie C., Deep recursive neural networks for compositionality in language. In *Advances in Neural Information Processing Systems* 2014. 2096-2104.

[4] Jia, Y. et al. 2014. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia* 2014. 675-678.

[5] Krizhevsky, A., Sutskever, I., and Hinton, G. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* 2012. 1097-1105.

[6] LeCun, Y., Bengio, Y., and Hinton, G. 2015. Deep learning. *Nature*. 521, 7553. 436-444.

[7] Lin, C.-Y. and Hovy, E. 2002. From single to multi-document summarization: a prototype system and its evaluation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (*ACL* '02). 457-464.

[8] Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*. 60, 2. 91-110.

[9] McDonald, R. 2007. *A study of global inference algorithms in multi-document summarization.* Springer Berlin Heidelberg. 557-564.

[10] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems* 2013. 3111-3119.

[11] Salakhutdinov, R., Mnih, A., and Hinton, G. 2007. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning* (*ICML* 2007). 791-798.

[12] Socher, R., Lin, C. C.-Y., Ng, A., and Manning, C. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th International Conference on Machine Learning* (*ICML*-11). 129-136.

[13] Van den Oord, A., Dieleman, S., and Schrauwen, B. 2013. Deep content-based music recommendation, In *Advances in Neural Information Processing Systems* 2013. 2643-2651.

[14] Xu, K., Ba, J., Kiros, R., Courville, A., Salakhutdinov, R., Zemel, R., and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of 32th International Conference on Machine Learning* (*ICML'15*).

Context-Aware User-Driven News Recommendation

Jon Espen Ingvaldsen
Norwegian University of Science and Technology, Department of Computer and Information Science, Trondheim Norway
jonespi@idi.ntnu.no

Özlem Özgöbek
Department of Computer Engineering, Balikesir University, Balikesir, Turkey
ozlem.ozgobek@balikesir.edu.tr

Jon Atle Gulla
Norwegian University of Science and Technology, Department of Computer and Information Science, Trondheim Norway
jag@idi.ntnu.no

ABSTRACT
Recommender systems match available contents with users' contexts and interests. With linked data knowledge bases we can build recommender systems where user interests, their context and available contents are modeled in terms of real world entities. In this demo paper we will describe existing academic news recommender systems and the Smartmedia prototype in particular. This prototype shows how we can combine available technologies like semantics, natural language processing and information retrieval to construct personalized and location aware recommendations on a continuous stream of news information.

Categories and Subject Descriptors
H.4.7 [**Information Systems Applications**] Communications Applications – *Information browsers*

General Terms
Algorithms, Design, Experimentation, Human Factors.

Keywords
Recommender system, news, mobile, natural language processing, named entity disambiguation.

1. INTRODUCTION
News organizations and libraries catalog their contents. These catalogs are traditionally constructed using controlled vocabularies with limited context information about what the catalog identifiers really mean. Even though a news article can be annotated with *"Barack Obama"*, there is no catalog information data saying he is the president of USA. The same article might be annotated with *"Boise"*, but we do not have data saying that it is a city and its longitude – latitude coordinates. Such extensive meta data attributes are valuable when we want to expose and personalize archive contents to a context aware user setting.

With large knowledge bases, such as WikiData and Yago, we get access to extensive databases of real world entities that are precisely described and structurally interlinked. By indexing news archives on such entity representations we can build news recommender systems that allow us to construct more ambitious catalog queries. For instance, we can retrieve all news articles from January 2015 related to the president of USA and locations within the range of 50 miles for the city center of Boise. This entity representation provides better solutions to challenges of news recommender systems like unstructured content, serendipty and synonmy [9].

The Smartmedia project[1] at NTNU targets construction of context aware news experiences based on deep understanding of text in continuous news streams [6, 11]. The goal of the Smartmedia project is to deliver a mobile and context aware news experience based on deep understanding of textual contents, combining both geo spatial exploration and context aware recommendations. In this project, we have implemented a prototype of a news recommender system where news articles are processed and represented in terms of WikiData entities. In this demo paper we will describe this news recommender prototype, its stream based data processing pipeline and its context-aware recommendation features.

Section 2 describe related work, followed by a description of the Smartmedia prototype. Section 3 describes how its stream based data processing pipeline is constructed and Section 4 presents its mobile user interface and interaction principles. Conclusions and future work is given in Section 5.

2. RELATED WORK
The prototype system, described in this paper, share similarities to other academic news applications such as NewsStand [10, 12] and News@Hand [2, 3]. Both these systems map textual news contents to entities defined in a knowledge base. NewsStand targets geo spatial exploration of news. It is an example application of a general framework developed to enable people to search for information using a map query interface. It utilizes maps both to explore and find news stories and to visualize and present single news events. News@hand combines textual features and collaborative information to make news recommendations. It uses Semantic Web technologies to describe the news contents and user preferences. Both news items and user profiles are represented in terms of concepts appearing in domain ontologies, and semantic relations among those concepts are exploited to enrich the above representations, and enhance recommendations. Both NewsStand and News@Hand have user interfaces targeting desktops and larger device screens. They both provide user control over the retrieved set of news, either through a map or category based navigation or preferences settings. The Smartmedia prototype targets mobile devices and combine some of the geospatial and ontological news data representation features found in both of two other system.

Asikin and Wörndl [1] propose news recommendation techniques based on a location aware context model. The spatial model proposed in this work contains geographical information like latitude and longitude, and a human described physical character of a location and place identity, which represents the location's meaning and significance. In this work it is also focused on the improvement of serendipity problem.

Building user profiles is an important aspect of recommender systems. Meguebli and Kacimi [8] propose an approach to build user profiles based on the comments added to news articles. To do that, entities are extracted from the user comments. Then the related aspects are extracted from the news articles. By using all the articles that a user read, a user profile is created based on the extracted aspects.

Another personalization approach in news recommender systems is Hermes framework [7]. In this work, in addition to the personalization an ontology based approach is used to recommend news articles. Also in [5] a semantic news recommendation framework which is called Athena is proposed. In this work, CF-IDF (Concept Frequency - Inverse Document Frequency) method is proposed which is the application of TF-IDF (Term Frequency - Inverse Document Frequency) method to semantic recommenders. CF-IDF is a selective method compared to TF-IDF where it considers only the key concepts in the news articles where TF-IDF considers all terms.

[1] http://research.idi.ntnu.no/SmartMedia

3. IMPLEMENTATION

The backend of the news recommender prototype developed is constructed as a pipeline of operations transforming Rich Site Summary (RSS) entries and raw text data into a semantic and searchable representation. The pipeline and its operations are implemented with using the Apache Storm[2] framework. This distributed computing framework enables scalability and ability to handle large amounts of news items from a magnitude of publishers continuously.

There are five steps involved in the data processing. The first step creates an input stream by continuously monitoring a set of RSS feeds from a wide range of news publishers. Whenever a new news item occurs, RSS entry properties such as the title, lead text and HTML sources are retrieved. The HTML sources are parsed and cleaned to extract a representative body text. In the second step, natural language processing operations such as language identification, sentence detection and part-of-speech tagging is applied to extract entity mentions from the textual data. The third step uses supervised models to map entity mentions to referent entities in the WikiData knowledge bases. These models combine textual similarities, WikiData graph relations and entity frequencies and co-occurrence statistics to classify the relevance of multiple referent candidates. First Story Detection is applied in the fourth step to group news items describing the same news story. In the fifth step this semantic representation is indexed and made searchable. As this backend architecture is stream based, it is able to index and promote recent news items soon after they are discovered.

WikiData is the community-created knowledge base of Wikipedia[13]. Since its public launch in 2012, the knowledge base has gathered more than 15 millions entities, including more than 34 million statements and over 80 million labels and descriptions in more than 350 languages[4]. Most geographical entities in WikiData provide a reference to Geonames containing more detailed geographical properties. In the implementation of the Smartmedia prototype, the entity information from these knowledge bases where indexed in a Lucene[3] based search index. This index makes the entities searchable and creates a foundation for addressing entity labels, descriptions and aliases, entity relations and geospatial properties.

Figure 1 shows an example of a news article from the Guardian where the text is parsed and enriched with WikiData entity annotations. The fields and nested data structure in this figure are similar to how the news stories are stored and indexed in the Lucene based index. By running the news text from the news article in the figure through the data processing pipeline, we identified nine WikiData entities, including Bedfordshire, Home Office and Theresa May. Note that the news texts and list of entities and associations in the figure is shortened. All entities contain a textual description and a list of associations. These associations are typed relations to other WikiData entities. We can see that Bedfordshire contains eight such entity associations. Examples of entities linked and related to Bedfordshire are the *instance of* relations to *Ceremonial county of England* and *Administrative territorial entity of the United Kingdom*. Both Bedfordshire and Home Office are additionally described with geospatial properties. In this case the geospatial properties are longitude – latitude pairs, but the implementation allows for any geo spatial shape decribed as valid Geojson[4].

When a user is opening the news app on the mobile a request containing user id, location and preferences are sent to the backend. Here, a multi factor search query is formed to retrieve relevant news entries from the index.

4. USER INTERFACE

A web-based and responsive user interface is developed to make the news stream contents explorable on mobile devices. In this interface, the user is allowed to extract news items that are relevant to the geo special locality context, personal interests and given point of time. These three relevance factors are customizable and the user can select whether or not they should influence the retrieved news items.

To customize the geographical locality, the user specifies a circular relevance region on a map. Figure 2a shows an example of such a relevance region. By default, the relevance region is set to users current GPS location with a 50 km radius. By moving the region or modifying the radius, users can generate a local newspaper for any region of the world. If the location factor is disabled, it means that the system is recommending news from any location in the world and news that are not containing location information.

In the current Smartmedia prototype, we have predefined a handful of user interest profiles. Each user profile contains an alias and a weighted vector of WikiData entities. Examples of predefined profiles in the system are stock trader, soccer fan, technology geek, etc. By selecting any of these interest profiles, the retrieved news will be influenced and biased towards the interest topics. When the personal interest factor is disabled, the user retrieve a news composition which is general and without such bias.

By changing the time-factor, the user is presented with a calendar where can move in time and retrieve either recent or historic news items. When, the time-factor is disabled the user will retrieve news solely based on the other relevance factors (location and personal interests).

Figure 2b shows an example of how news stories are presented. Here we see the same article as we had in Figure 1. The three circular buttons on the bottom of the screen allow users to toggle whether their locality, personal interest profile and time setting such influence news story retrieval.

By clicking on a news story, the user gets the ingress of the news story and a list of the most salient entities for the selected news story. Figure 2c shows the ingress and relevant WikiData entities from the news article about Theresa May. As we can see, our news story about politics and terror related to Syria, Theresa May, ISIL and Sky News. By hovering these items, the user is presented with their textual WikiData description. On figure 1c, we can see that the WikiData entity for Theresa May contains the description *"British politician"*.

In general, the three buttons at the bottom of the screen for location, interest profile and time can at any time be activated and de-activated in combinations to provide very different recommendation strategies. For example, keeping all buttons active with default parameters means that the system will recommend news articles that have recently takes place in the vicinity of the reader and are consistent with her profile. A screencast video describing the features of the system and its user interface is available at https://vimeo.com/121835936

5. CONCLUSIONS AND FUTURE WORK

Many see the full stack of semantic web technologies as a complex implementation of some really simple and good ideas about adding meaning to data. There are great rewards in understanding the full stack and what it can do, but most news organizations find great rewards by looking into linked data in combination with traditional information retrieval techniques.

In this paper we have shown a prototype of a news recommender system that demonstrates some of the context and geo spatial aware features online news services can achieve by using available and open knowledge bases and data processing and storage technologies.

Future work for the Smartmedia prototype will focus on improvement on entity linking qualities and evaluations of user needs. The user evaluations will look into to which extent users find the ability to control their news feed in terms of location, interest profile and time valuable and useful.

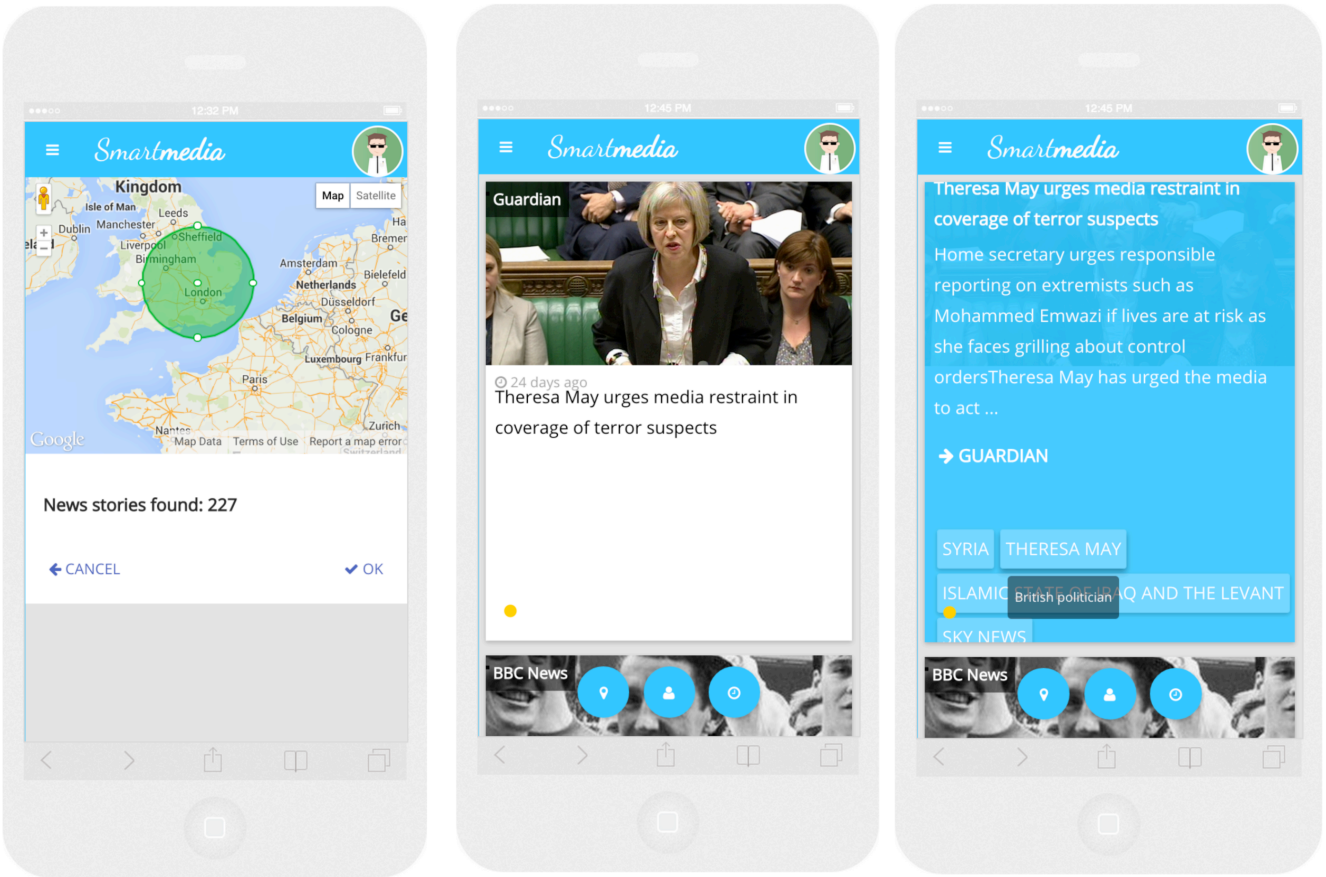
[2] http://storm.apache.org/

[3] https://lucene.apache.org/core/

[4] http://geojson.org/

```
articleId: "Guardian_254439378"
type: "article"
title: "Theresa May 'allowed state-sanctioned abuse of women' at Yarl's Wood"
leadText: "Shadow home secretary criticises minister after TV documentary alleges rape and self-harm at detention centre were
ignoredTheresa May, the home secretary, has been accused of allowing the "state-sponsored abuse of women" at the Yarl's Wood detention
centre after a Channel 4 investigation uncovered guards ignoring self-harm and referring to inmates in racist terms.Yvette Cooper..."
entities:  [ 9]
          0:   {
                    entityId: "Q23143"
                    name: "Bedfordshire"
                    description: "county in England"
                    associations:  [ ... 8]
                    shape:  {
                              type: "Point"
                              coordinates:  [ 2]
                                        0:  -0.41666666666667
                                        1:  52.083333333333
                    }
          }
          1:   {
                    entityId: "Q763388"
                    name: "Home Office"
                    description: "ministerial department of the Government of the United Kingdom"
                    associations:  [ ... 3]
                    shape:  {
                              type: "Point"
                              coordinates:  [ 2]
                                        0:  -0.129948
                                        1:  51.4958
                    }
          }
          2:   {
                    entityId: "Q264766"
                    name: "Theresa May"
                    description: "British politician"
                    associations:  [ ... 21]
          }
```


Figure 1. Example of a news article enriched with WikiData entities.



a) b) c)

Figure 2. Screenshots from the Smartmedia prototype. a) The map query interface. b) Presentation of news stories. c) Presentation of news details.

6. REFERENCES

[1] Asikin, Y. and Wörndl, W. 2014. Stories around You: Location-based Serendipitous Recommendation of News Articles. *Proceedings of 2nd International Workshop on News Recommendation and Analytics*. (2014).

[2] Cantador, I., Bellogín, A. and Castells, P. 2008. News@ hand: A semantic web approach to recommending news. *Adaptive hypermedia and adaptive web-based systems*. (2008).

[3] Cantador, I., Bellogín, A. and Castells, P. 2008. Ontology-based personalised and context-aware recommendations of news items. *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. 1, (2008).

[4] Erxleben, F., Günther, M. and Krötzsch, M. 2014. Introducing Wikidata to the Linked Data Web. *The Semantic Web–ISWC 2014*. (2014).

[5] Goossen, F. and IJntema, W. 2011. News personalization using the CF-IDF semantic recommender. *Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS)*. (2011).

[6] Gulla, J.A., Ingvaldsen, J.E., Fidjestøl, A.D., Nilsen, J.E., Haugen, K.R. and Su, X. 2013. Learning User Profiles in Mobile News Recommendation. *Journal of Print and Media Technology Research*. II, 3 (2013), 183–194.

[7] IJntema, W. and Goossen, F. 2010. Ontology-based news recommendation. *Proceedings of the 2010 EDBT/ICDT Workshops*. (2010).

[8] Meguebli, Y. and Kacimi, M. 2014. Building rich user profiles for personalized news recommendation. *Proceedings of 2nd International Workshop on News Recommendation and Analytics*. (2014).

[9] Ozgobek, O., Gulla, J. and Erdur, R. 2014. A survey on challenges and methods in news recommendation. *In Proceedings of the 10th International Conference on Web Information System and Technologies (WEBIST 2014)*. (2014).

[10] Samet, H., Sankaranarayanan, J., Lieberman, M.D., Adelfio, M.D., Fruin, B.C., Lotkowski, J.M., Panozzo, D., Sperling, J. and Teitler, B.E. 2014. Reading news with maps by exploiting spatial synonyms. *Communications of the ACM*. 57, 10 (Sep. 2014), 64–77.

[11] Tavakolifard, M., Gulla, J.A., Almeroth, K.C., Ingvaldesn, J.E., Nygreen, G. and Berg, E. 2013. Tailored news in the palm of your hand: a multi-perspective transparent approach to news recommendation. *WWW '13 Companion Proceedings of the 22nd International Conference on World Wide Web*. (May 2013), 305–308.

[12] Teitler, B. and Lieberman, M. 2008. NewsStand: A new view on news. *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*. (2008).

[13] Vrandečić, D. and Krötzsch, M. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*. (2014).