Social Networks Serving Web Feeds:

An Approach for Web Feed Enrichment

Abstract

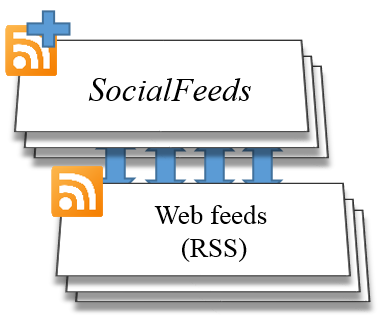
A web feed is a new kind of web services that was created in the past two decades to keep remote users updated with the latest news without having to crawl individual web sites. Currently, the existing web feed services get their information from one source (the website hosting the feed) in a standardized format that is normally structured as follows: title, description, link, image and date. In this paper, we propose a new approach called NERVES (social Networks sERVing wEb feedS) that connects web feeds to a Social Networking Sites (SNS), and aims at (1) keeping users aware of the wider context taking into account both the source of the feed and the SNS. (2) Our approach enriches existing feeds with extra information harvested from the social web. We validated the efficiency and accuracy of our approach on public data and report on empirical results yielding an accuracy of 94%.

1. Introduction

A user typically accesses browser feeds through a feed reader that can be a client software embedded inside a browser agent or a web application, to be notified automatically of the recent news. The users often subscribe with specific websites based on their interests. The feeds normally (RSS, Atom, etc.) composed of a light content; a title, URL, image and content. Although websites are still providing RSS services for users, actual users’ count has seen a severe incline and the service is losing people engagement. This paper aims at enhancing the value of web feeds by connecting them to the SNS and correlating relevant information such as text, and multimedia from these SNS to make the feed content wealthy and more interactive.

By connecting web feeds to such SNS, the feed can be enriched on two spheres. (1) by structure we can extend the current attributes, for instance, we can add URL, image or videos to the feed that are missing from the original feed. (2) by content, we can enrich the existing content with information collected from SNS adding more representative textual content (title, description) or multimedia (image, video) as well as URIs that allows users explore more external resources (links) of the feeds.

In this paper, we addressed the problem of web feeds enrichment to keep users aware of a wider context of information fused from both the source of feed and the SNS. The contributions of this work are as follows: (1) Enriching the web feeds by structure and content. (2) Development of a linking algorithm that connects web feeds to SNS. (3) Making NERVES as an adaptive and flexible web plugin that works on the top of any web feed service as shown in figure 1.



*NERVES*

*Web feeds*

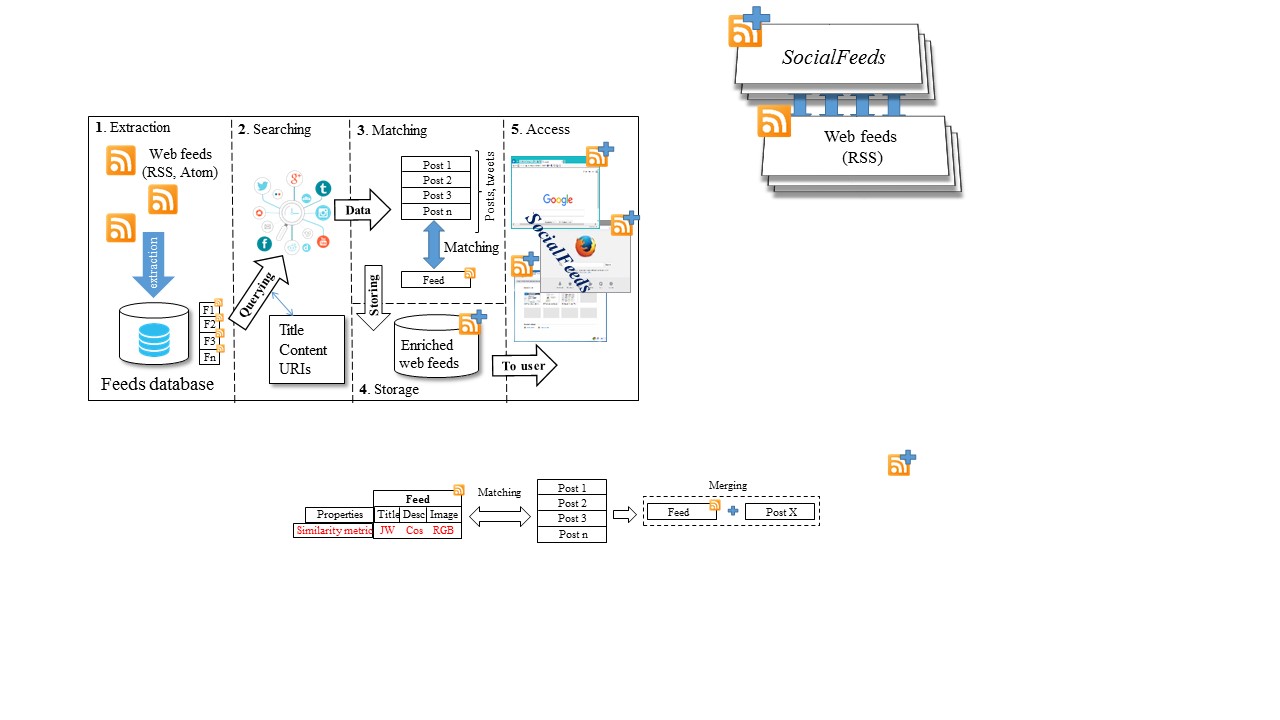
Figure 1: NERVES on the Top of the Original Web Feeds

The rest of this paper is organized as follows: in section two we discuss the related work in this field, in section three we display our model and the solution and in the last section we show the experimental results and evaluation of our approach.

1. Related Work

Hurtado [1] guides a web crawler by resultant RSS feed to fetch information from articles. They benefit of the chronological order of the RSS feed to ease the work of the web crawler and to make it more efficient. They extract data from raw HTML pages by identifying the HTML tag containing news content by the number of characters in a tag, then stripping the HTML tags to retrieve the text corpus. The extracted text corpus is processed by NLTK to find the most relevant keywords and bigrams and eliminate stop words. In addition, the image with the highest area is selected as the main image, if none exists or the quality is low, public image databases are searched for suitable images.

In another contribution, Dhahri et al. [2] manage to mine events gathered from Social Media RSS feed in two websites specialized in social events. The steps were as follows, first after gathering an RSS feed stops words are eliminated, then construct a bag words, then select features for model learning. Finally, they execute a learning model based on a set of equations, which is capable now to identify the appropriate event.

Saha et al. [3] process news items in RSS feeds in a Web services using text categorization techniques and then delivers categorized news items to a client application. The client application requests the Web service to provide specific news items base on predefined categories.

Han et al. [4] extract article content from news web pages based on RSS feeds. The automated system uses the feed title to get a keyword list then detect the position of news title in the news page deduced from feed’s URL. Then paragraphs of the news article are detected and extracted as full text after detecting the news title on the news page.

Cobra (Content-Based RSS Aggregator) by Rose et al. [5] is a system that delivers users personalized feeds based on their interests. The system uses a set of crawlers, filters, and aggregators to accomplish the task.

In another contribution, Araibi et al. [6] introduce refined RSS recommendations base on a data warehouse perspective after considering a set of RSS as multi-dimensional data stored in a data warehouse.

Rao et al [7] propose an architecture to overcome the high bandwidth consumption issue, and limited filtering semantics in RSS feeds/readers architecture. The formerly mentioned architecture lets users subscribe to favorite contents via input keywords. Also, cooperative content polling, filtering and disseminating via DHT-based P2P overlay networks save network bandwidth consumption.

According to the aforementioned systems and approaches, the web feeds enrichment problem has not been addressed yet. Furthermore, the social media is miss-used in any web feed service. Hence, we contribute in this paper in making feeds more lovely and interactive for end user, by leveraging the wealthy content of social networking services for enriching purposes.

1. Approach Overview

In this section, we illustrate and describe the overall architecture of NERVES. The components of the architecture are described briefly at the beginning, and then we explain each component in separately in details.

Figure 2 shows the architecture of our NERVES system. The system mainly consists of five main components:

**1.** Extraction: it extracts feeds from the web automatically. It first collects and identifies a feed resource on the web, then uses extraction methods to extract the feed properties from the identified resources.

**2.** Searching: based on the extracted properties, it searches different social networking platforms for similar contextual data.

**3.** Matching: it matches the results of the searching component, that could be tweets, posts (from Facebook pages and/or from private user accounts) with the corresponding feed using a matching algorithm

Figure 2: NERVES Architecture

**4.** Storage: it stores the matched and enriched web feeds (by information from social networking sites) in a specific database, specifically it employs MySQL for storage.

**5.** Access: it provides a new service for browser agents on top of current web feed services. Those are enriched with extra content from social networking sites. It also provides access to the list of URLs of the connected feeds on the SNS.

3.1 Extracting and Searching

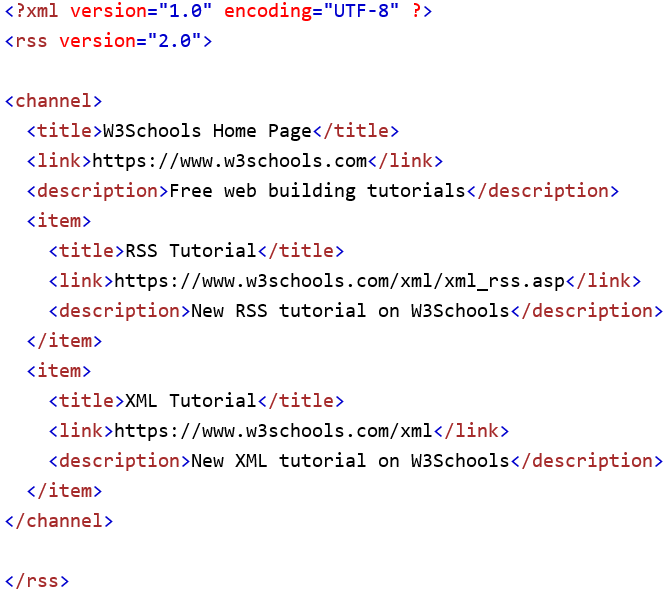
The extraction process starts by defining a list of feeds originating from a feed service such as RSS or Atom. In this paper, we conduct all of our experiments on RSS feeds. First, the feeds are parsed from the channels, which is usually XML structured anatomy as shown in figure 3.

Figure 3: An Example of an RSS Feed with XML Structure

Then we process the textual contents (title and description) as follows:

Stop words removal: the stop words are removed from the title and description to overcome the interruption of searching efficiency. The stop words were pre-defined in a special dataset and each feed is matched on this set to detect the existence of stop words and remove them.

Special characters removal: the special characters are removed from the feed title and description as well. All the special characters have been defined in a specific regular expression (RegEx) and each feed is matched on this RegEx to detect the existence of special characters and remove them. Special character, e.g. exclamation marks and commas.

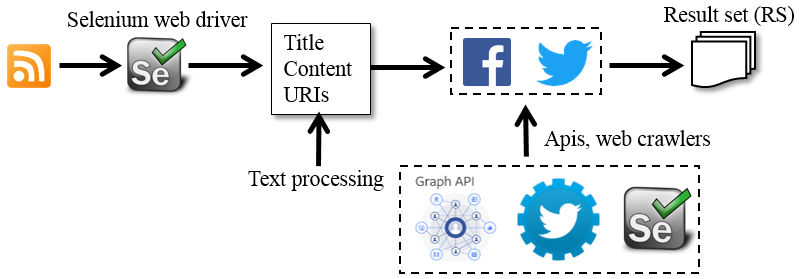
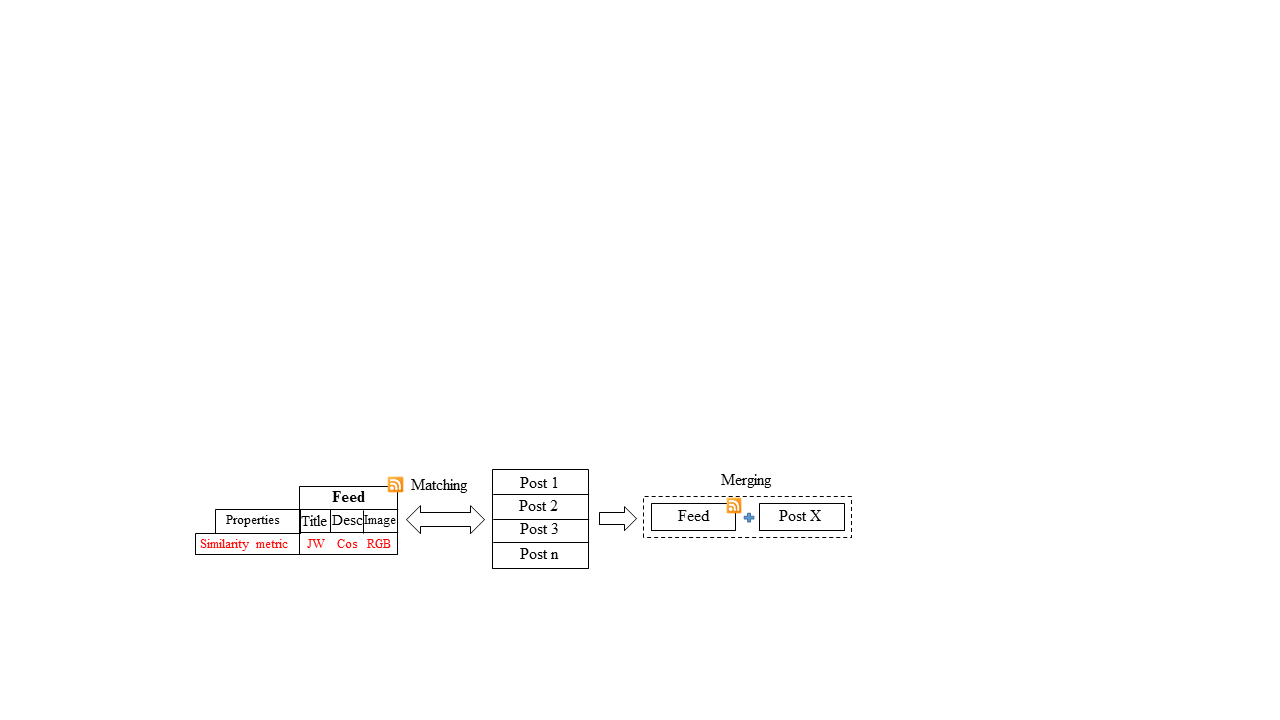
To crawl data from the SNS, we have used the Selenium web driver [9] that automates the extraction by opening a connection from Selenium to the inspected elements in the web pages inside browsers. In addition, we have used Facebook Graph api that allows the extraction of any data from Facebook pages and the Twitter api as well for the same job.

Figure 4: Information Extraction Module

Each SNS is queried with information extracted from the feed properties, as illustrated in our data extraction module in figure 4. We used two queries: the title and the description (after processing), each query returns a Result Set (RS); this set is matched with the corresponding feed using specific similarity functions and matching algorithm.

3.2 Matching

The corresponding matching candidates (posts, tweets) from each feed are those who retrieved in the searching component. For each feed property, we employed its suitable similarity metric. In figure 5, we illustrate this taxonomy in three steps assigning attribute/similarity functions, matching and finally merging. Jaro-Winkler (JW) is used to match strings inside titles [8]. The cosine similarity (cos) (1) is used to match long string tokens as paragraphs [10] as well as the description in the feed. A and B are two descriptions from feed and SNS. The RGB Histogram (RGB) is used to compare to images if they are similar to each others.

The matching algorithm starts by connecting each feed with its corresponding post(s)/tweet(s) from such SNS. After we did comprehensive overview we conclude that feeds can be matched mostly by titles/keywords in a title using the Jaro-Winkler similarity. The second matchable attribute is the description due to the abundance of the content. This is matched if the matching result returned by title comparisons is zero. It uses the cosine similarity as mentioned before. Finally, if both title and description return zero we try to find matched content on such SNS using the image comparison, for this task, we employed the RGB Histogram image similarity metric. At the end of the matching process, we couple the original RSS feed with the corresponding post(s)/tweet(s) available on SNS providing for each feed

Figure 5: Matching Module

a list of the URIs of the matched content on SNS. This allows the end users to get access to the external contents as well.

Algorithm 1 shows a brief pseudo code that states the inputs, output and the method of matching. We assign a threshold T for each similarity function. Each threshold is calculated based on the following mechanism:

* 100 matching candidates have been calculated on all of the three similarity metrics, the threshold is obtained by calculating the average score of the truly matched pairs. Tx (2) Is the threshold function where x is the similarity function (Jaro-Winkler, Cosine similarity, RGB Histogram), n is the number of truly matched candidates and N is the total number of matchings (100).

The threshold function can be obtained also using machine learning algorithm, so it can learn from the average score and adopt the threshold value.

**Algorithm 1**: Deciding the matched candidate

**Input:**

Feed: single web feed (RSS)

SNS-RS: a set of posts, tweets from SNS

TJW: The threshold Jaro-Winkler

TCos: The threshold of cosine similarity

TRGB The threshold of RGB Histogram

**Output**:

Coupled feeds

1. **begin**
2. **foreach** SNS-RSiin SNS-RS do
3. **if** (Match(SNS-RSi.title, Feed.title)<= TJW) then
4. **if** (Match(SNS-RSi.description, Feed.description)>= TCos)
5. Couple (SNS-RSi. Feed)
6. **Else**
7. **//**Match by image
8. **If** (Match(SNS-RSi.image, Feed.Image) >= TRGB)
9. Couple (SNS-RSi, Feed)
10. **end**
11. **end**
12. **end**



In Table I, we present the average similarity scores of Jaro and cosine similarity on different topics. We can observe that sports, for instance, has the highest scores. This is due to the similarity of this topic content, for instance: scores, winner names, tournament location, etc.

Table 1: Similarity Scores Using Default Similarity Metrics

|  |  |  |
| --- | --- | --- |
| **Feed topic** | **Attribute/Similarity** | |
| **Title** | **Description** |
| **Jaro** | **Cosine** |
| *Business* | 0.86 | 0.52 |
| *Sports* | 0.94 | 0.76 |
| *Politics* | 0.78 | 0.42 |

3.3 Enrichment

The matching process matches each single web feed in the feed set with similar feeds existing on a such social network, using the defined similarity metric. Each feed *F* is matched with one (*F*↔*Fs*) or many similar feeds (*F*↔), where *Fs* is the feed from a social network and *n* is the number of feed.

The enrichment process is composed of two phases: (1) the filtering and (2) the merging phase. Phase two process is responsible on omitting the similar records in set before merging them.

(1) Filtering

Each item in contains four items: Title, Description, Image and Url ←{Tn,Dn,In,Un}, each record is compared to others using these four items.

Table 2: An Example of Comparing feeds from social networks

|  |  |
| --- | --- |
| Feed | Similar items |
| ← {T1,D1,I1,U1} | U1=U3  I1≠I2≠I3  U2≠{U1,U3}  D1=D2=D3  T1=T2=T3 |
| ← {T2,D2,I2,U2} |
| ← {T3,D3,I3,U3} |

In table 2, we illustrate an example of three feeds from social networks, each feed is composed of four items. The similar items are detected among this set as shown in column 2. We derive that URLs from feeds one and two are the same, all the images are different, all descriptions and titles are the same.

(2) Merging

All feeds in the set are merged according to the filtering facts in the previous phase, using the merging function (3). contains distinct items as referred in Table 2.

The feed *Fnew* is enriched then with the following content:

1. One new description
2. Two new URLs
3. Three new images
4. One new title

Following the enrichment process, the feeds are stored in a local repository for later access as described in the next section.

3.4 Storage and Access

All the enriched feeds in our experiments have been stored in a MySQL database as well as for the end user service, the enriched feeds are stored in this database and after we bring them to the user. As we mentioned in the introduction, this tool is flexible. The user can adapt the following functionalities:

**1.** The resource(s) he wants to use (Facebook and/or Twitter)

**2.** The attribute(s) he wants to enrich (Title, Description and Image).

**3.** Enriching the feed or just connecting it with to the SNS by providing a list of URLs of the matching candidates.

The user after installing NERVES as an extension of a browser agent, he must define the following inputs:

**1.** The resource of the RSS feed service (a URL)

**2.** If he prefers to specify the SNS URLs, he can enter a list of Facebook, Twitter page(s) links.

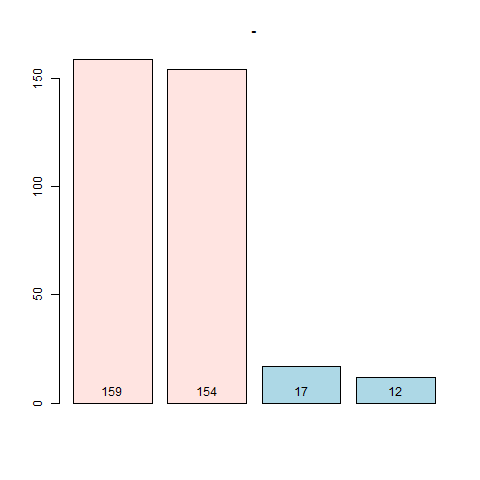
3. Similarity functions are selected by default.

1. Experiments and Evaluation

We divide the analysis in this section for two parts, firstly, we overview the dataset used, and the experiments have been conducted. Secondly, we show the metrics to evaluate the performance of our approach.

4.1 Dataset and Experiments

In our experiments, we used two social networking sites; Facebook and Twitter. The tests are conducted on RSS web feeds only, due to its popularity. The dataset we used composed of a set of feed sources that Reuters use to send news to users of total of 188 feed. These sources are composed of a set of Category/URL users can copy to feed aggregators to benefit from the Reuters RSS feeds service. An e.g. of such category in Reuters set arts are business and health news. Reuters feed sources are available free of charge, for individual and non-commercial use, and can be found on “reuters.com/tools/rss”. For each feed we get a list of properties (title, description, link and image if exists).



Enriched with

Text

Enriched with

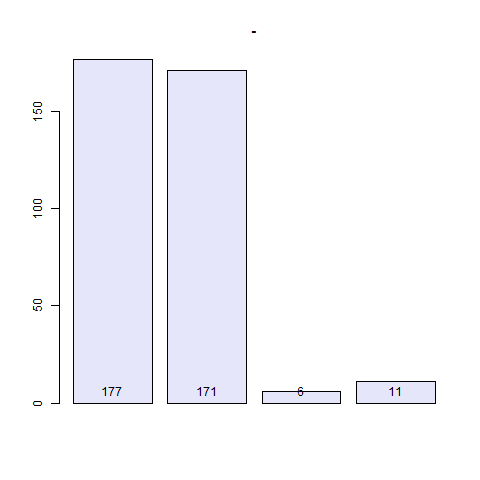
Text

Enriched with

Image

Enriched with

Image

Figure 6: Distribution of Feeds Matched by Content and Attributes

Enriched with

Text

Unmatched

Feeds

False

Matched

Feeds

Truly

Matched

Feeds

Total

Matched

Feeds

Figure 7: True/False Matches

Figure 6 represents the number of feeds that have been enriched by content (first two bars) on two different attributes, text (title, content) and images. The second two bars represent the total numbers of feeds have been enriched by structure using the same attributes mentioned before. The feeds as observed have enriched by content more than structure.

Figure 7 shows that the total unmatched feeds are 11 feeds representing only 6.2% of the total matched feeds. The overall accuracy of our system is 94.1%.

Table 3: The Contribution of Each Social Networking Site in the Enrichment of Each Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Enriched by content** | | **Enriched by structure** | |
| **SNS** | | **SNS** | |
|  | **Facebook** | **Twitter** | **Facebook** | **Twitter** |
| Title | 47% | 53% | 63% | 37% |
| Description | 69% | 31% | 35% | 65% |
| Image | 84% | 16% | 75% | 25% |
| URIs | - | - | 32% | 68% |

Each social networking site contributes differently in the enrichment according to the information existed on it, the volume can be different as well as content. Hence, we studied the plenty of each attribute on Facebook and Twitter and the results are displayed in table II. Results shown for the content contribution, that description (long texts) exists highly on Facebook as titles (short texts) exists highly on Twitter. Concerning the structure enrichment, Twitter had contributed mostly by enriching the feed structure by description, due to the massive existence of Tweets on Twitter than Facebook. The same is with URLs, they considered as textual properties that is founded inside the Tweet content. Existence of images on Twitter are mostly come from embedded images inside external shared links. However, on Facebook images have been found in the real post.

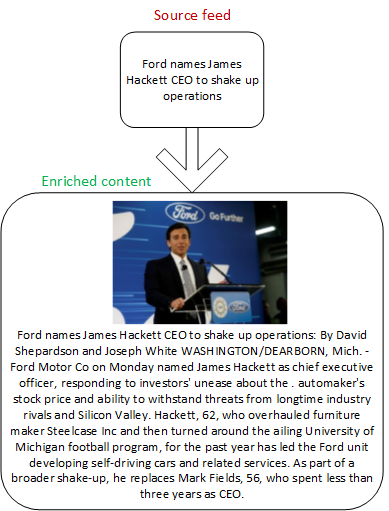


Figure 8: Example of an Enriched Feed

In figure 8, we have an example of enriched RSS web feed with both content and structure. This feed is enriched with additional textual content (description) and with additional attributes (new image). The feed is being lovelier to the end user and more interactive which might raise his motivation to engage web feeds services.

4.2 Performance Evaluation

We evaluate our approach on two evaluation metrics; the precision R (4) and recall R (5). Both metrics are evaluated on Facebook and Twitter SNS. They have been evaluated on different numbers of feeds.

The precision has highest values than recall on Twitter compared to Facebook, this is because Twitter content is less complex than Facebook. Deeply speaking, Facebook content is more social than other SNS, many things can be shared such as, images, large text, life events, activities, etc.

|  |  |
| --- | --- |
| .2 .4 .6 .8 1 | .2 .4 .6 .8 1 |

Figure 9: Precision Chart Figure 10: Recall Chart

On the other hand, Twitter is restricted for short tweets that contain only textual data, in addition to the sharing possibility of external links.

1. Conclusion and Future Work

In this paper, we present NERVES, a tool for connecting and linking RSS feeds to Social networking sites. We aim at making web feeds lovable to users and more interactive. To achieve this goal, we build a matching algorithm that connects each unique feed with the similar information on SNS and merge them. The feed then is enriched with both content and attributes. As a future work, we aim at making this approach more popular and add more functionalities, so it can be accessed as a real new version of the current web feeds. In addition, current tests have been evaluated against the political dataset, as little evaluations on two other topics were performed. Thus, we will work and including more diverse dataset topics, so we can have a more trusted evaluation of the tool performance.

References

Clancey, W. J. 1984. Classification Problem Solving. In *Proceedings of the Fourth National Conference on Artifi­cial Intelligence,* 49-54. Menlo Park, Calif.: AAAI Press.

J. Hurtado, "Automated System for Improving RSS Feeds Data Quality," CoRR, vol. abs/1504.01433, 2015.

N. Dhahri, C. Trabelsi and S. Ben Yahia, "RssE-Miner: A New Approach for Efficient Events Mining from Social Media RSS Feeds.," in DaWaK, 2012, pp. 253-264.

S. Saha, A. Sajjanhar, S. Gao, R. Dew and Y. Zhao, "Delivering Categorized News Items Using RSS Feeds and Web Services," in 2010 10th IEEE International Conference on Computer and Information Technology, Bradford, 2010, pp. 698-702.

H. Han, T. Noro and T. Tokuda, "An automatic web news article contents extraction system based on RSS feeds," Journal of Web Engineering, vol. 8, no. 3, pp. 268-284, 2009.

I. Rose, R. Murty, P. Pietzuch, J. Ledlie, M. Roussopoulos and M. Welsh, "Cobra: Content-based Filtering and Aggregation of Blogs and RSS Feeds," in 4th Symposium on Networked Systems Design and Implementation, 2007.

N. Araibi, E. Ben Ahmed and W. Karaa Ben Abdessalem, "IRORS: intelligent recommendation of RSS feeds," Vietnam Journal of Computer Science, vol. 3, no. 1, pp. 47-56, 2016.

W. Rao and L. Chen, "A distributed full-text top-k document dissemination system in distributed hash tables," World Wide Web, vol. 14, pp. 545-572, 2011.

W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. In Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03).

10 20 30 40 50

10 20 30 40 50

http://docs.seleniumhq.org/

Singhal, Amit. "Modern Information Retrieval: A Brief Overview". Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24 (4): 35–43.