FOOD & BIOBASED RESEARCH
WAGENINGEN UR

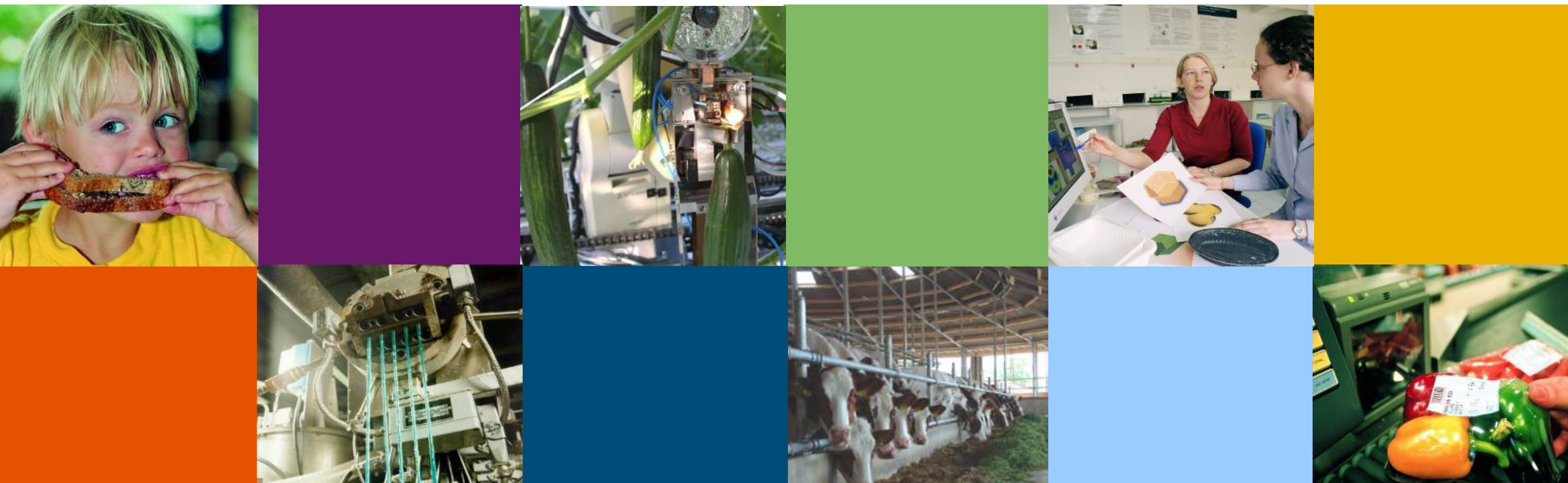
Social Media

Embedding in the eFoodLab Framework

Don Willems and Jan Top

COMMIT/

Report 1532



Colophon

Title	Social Media
Author(s)	Don Willems and Jan Top
Number	1532
ISBN-number	978-94-6257-396-3
Date of publication	29 January 2015
Confidentiality	No
OPD code	
Approved by	Nicole Koenderink

Wageningen UR Food & Biobased Research
P.O. Box 17
NL-6700 AA Wageningen
Tel: +31 (0)317 480 084
E-mail: info.fbr@wur.nl
Internet: www.wur.nl

Abstract

The data generated by users of Social Media is enormous. For computers to be able to use this information, which is often in textual form only understandable by humans, meaning (semantics) has to be added to the information. Because of the enormous amount of information this has to be done automatically by computer systems that are able to parse the textual information and add semantic annotations. In this report the software components that were developed by the Information Management Group at Food & Biobased Research to enable automatic annotation of Social Media messages are described.

Content

1 Introduction

The information created by users in different Social Media can be relevant for many different applications such as decision support systems. Unfortunately the sheer size of the information created in Social Media is a major obstacle for using this information, which is often in textual form meant for human consumption. Automatically adding meaning (semantics) to this information will enable computer system to use the information (data) provided by Social Media. This report describes the software components that were developed within the KB project "Social media and social capital of online social networks" (KB 16-001.06-005) and embedded in the semantic search and annotation framework developed in the eFoodLab COMMIT/ project. These software components allow for the scheduled downloading, automatic annotation, and indexing of social media messages using the functionalities provided by the existing smart search and annotation components developed within the eFoodLab project. Annotation and indexing is done relative to domain ontologies specified by either the developer of a specific implementation of this software or optionally by the user if needed.

For a demonstration given a the COMMIT/ "The Big Future of Data" event a semantic search demo was given in which users could search through a large set of documents from the University Library Wageningen using specific domain ontologies for for instance policy or ICT. If the user started typing "social me" into the search field, he or she was presented with a list of possible completions for the search term, such as "**social me**dia". Social Media is a concept in the ICT ontology. When the user starts the search, the search index is queried for the concept "Social Media" and any existing connections to specific documents or texts are returned as search results. The search index contains links between documents and the concepts they contain, the specific text of the occurrence of the concept in the text and the number of occurrences in the text. This allows for very efficient search for the concepts specific to a domain (as defined by the domain ontology). The Social Media components presented in this report add this functionality specific for social media messages.

2 Social Media

Social Media are defined as "forms of electronic communication through which users create online communities to share information, ideas, personal messages, and other content" [1]. This includes, but is not restricted to social networking [2] where users are also able to "articulate a list of users with whom they share a connection" [2] and view and traverse their (and other's) lists of connections [2]. While much research is conducted into understanding the nature of online social networks, our focus is on the content of the information shared using social media. Connections such as mentions in tweets are relevant in the sense that they provide links to new content and therefore information that may be relevant for our application domain.

As we are more interested in the content conveyed using social media than in the social network per se, other social media than the 'usual suspects' (e.g. Facebook, Google+, Twitter, Pinterest,…) may be more relevant.

2.1 Supported Types of Social Media

The smart search and (semi-)automatic annotation software developed within the eFoodLab project allow the semantic annotation of textual documents or fragments of documents. Semantic annotation has focussed mostly on webpages and PDF documents. Within the "Social Media and social capital of online social networks" KB16 project at Wageningen UR/Food & Biobased Research, components have been developed to include social media into the semantic annotation framework. We focussed on the most relevant social media for our domain.
With respect to content, the most important social media framework is one of the oldest; RSS-feeds. RDF Site Summary or, alternatively, Really Simple Syndication (RSS) is a (family of) web formats to publish frequently updated information such as published in news outlets, but also in personal blogs [3]. Although RSS-feeds are not always considered as part of Social Media [4], they do allow for the sharing of information. RSS-feeds however allow the user to aggregate pertinent information together for easier access [5] and linking using the user's own blog or other social media. Users of an RSS-feed can subscribe to a multitude of feeds to easily check for new items on a regular basis. Aggregators such as Feedly allow users to collect information from different sources and republish the information using social media. An important argument in favour of including RSS feeds is that users' trust in blogs is higher than in other social media. This is an important indication that the content of RSS feeds and blogs is more accurate.
We have also created Twitter interfaces to automatically download Tweets and the content linked from those tweets. While the information included in the tweets themselves is very limited due to the restriction in the number of characters, tweets often contain links with relevant content. The @mentions in tweets are used to find similar content.

The content of RSS feeds and tweets, and linked documents is automatically annotated using the existing semantic annotation software developed in the eFoodLab project. The annotations are then added to a semantic index allowing for fast and accurate retrieval of requested information.

The software (Application Programming Interface) was created in such a way that other social media platforms can easily be added to the software application. The content of those social media messages is then also annotated and indexed using the semantic annotation tools.

2.2 Automatic and semi-automatic annotation using ontologies

When a social media message (or linked website or PDF file is added to a search index using the eFoodLab Smart Search API, the content is parsed and analysed for metadata and for occurrences of ontological concepts within the text. The metadata being extracted contains information such as biographical data (title, publication date, authors, type of publication) and more specific metadata pertaining to social media. The bibliographic metadata is described using standard ontologies such as DublicCore (DC, [7]), Friend Of A Friend (FOAF, [8]), and the Bibliographic ontology (BIBO, [9]). Metadata specific to Social Media is included using ontologies such as the Semantically-Interlinked Communities ontology (SIOC, [10]) and the RSS 1.0 [11] ontology.

After bibliographic metadata has been extracted, the content text is searched for occurrences of (the labels of) the different concepts in selected domain ontologies. The textual content of the document and of the ontology is lemmatised to account for difference because of for instance plurality.

All metadata and annotations are stored in an RDF triple store as RDF. The URI used to identify the document (blog post, RSS feed item, or Tweet) is either the URI initially associated with the document (in the case of RSS feed items that use an URI), the URL of the document that is the web address at which the document can be found, or a constructed URI when no URI or URL can be associated with the document. This constructed URI contains a randomly created UUID as the local name in the URI.

The annotations stored in the triple store can be used as a semantic index used in our semantic search application toolbox. Concepts from a domain ontology are linked to occurrences of (the labels of) the concept in the documents that have been annotated and added to the semantic index. Searching for a specific concept (or text describing the concept) then simply involves selection of the concept to be searched for, following the edges within the RDF graph of the semantic index to the documents in which the concept occurs and then returning a description (using for instance the metadata associated with the document) of the document.

3 Embedding in the existing eFoodLab Framework

The Social Media components are designed so that they can easily be reused in different (web) applications/services.
They are integrated in the Pipelines architecture that is used at Food & Biobased Research to easily create custom web services. They can also be modified easily allowing for different use cases.

They are also integrated into existing software components for semantic search and annotation. The developer whishing to use these components may elect to use the pipelines architecture to create web services but may also elect to use the social media components separated from the pipelines architecture depending on the requirements for the software application being developed.

3.1 Integration into the Pipelines architecture

The Social Media monitoring web services are fully integrated into the Piplines framework used at Food & Biobased Research for easy web service development (see [12] for details). Pipelines use three different processors: i) Generators to access the data needed for the web service to perform its tasks, ii) Transformers to transform the data into the correct output format, and iii) Serialisers to serialise the data in the correct format and adjust the HTTP response accordingly. For the Social Media web services, only one generator for each social media framework is needed. As only RSS-feeds and Twitter are currently supported, only the `FeedAdministrationGenerator` and `TwitterAdministrationGenerator` were needed. These generators support administration tasks such as adding new feeds (using their URL) and adding twitter users to be followed and adding queries on Twitter to be executed. These generators need configuration parameters such as the Twitter account used for searching and consumer and access tokens to be specified in the `pipelines.xml` configuration file. The developer can specify multiple pipelines for different Twitter accounts if needed. Accessing the annotations and searching feeds and tweets is supported by the existing smart-search and annotation frameworks that are also integrated with the Pipelines architecture. These pipeline components can easily be connected together to create custom web services and can easily be reused.

3.2 Semantic annotation and search

Possible outputs of the social media components are the metadata pertaining to the document/message/tweet and the textual content of the document/message/tweet. To perform automatic annotation on the content is as simple as adding the textual content, metadata and selected domain ontologies (or references/URLs to these ontologies) to a semantic index. This semantic index has been developed previously within the eFoodLab project to be able to search using semantic concepts within PDF documents and websites (HTML pages).

4 Application Programming Interface (API)

The application programming interface (API, see Figure 1.) for the Social Media components consists of only a few classes. The core interface is `ScheduledSmartSearchTask`, which defines two methods for task scheduling specific for Social Media monitoring. The `runScheduledTask` method is run each time an internal timer with an interval defined by the developer or user fires. For each supported Social Media platform a class is defined that implements the `ScheduledSmartSearchTask` interface. Each of the implemented Social Media platforms is queried for new messages each time the timer fires.
RSS feeds are implemented in the `FeedManager` class. Users/developers can add feeds (more specifically the URL at which the feed can be found) to the feed manager. Users can also update feeds using the `updateFeed` methods. The `updateFeed` methods can be invoked programmatically by the user or are invoked when the timer fires. While updates can always be done, an actual download of the feed will only take place when the update does not violate the update policy of the feed. If the feed administrator sets the update policy to daily, only one update will take place per day. The user, however, is able to force the update but that is not recommended.

The `TwitterManager` can be used to download tweets from specific users or use predefined queries on twitter. The update methods in this manager can be used to manually update the twitter messages or is invoked by the timer.

Both the `FeedManager` and the `TwitterManager` have a reference to a semantic search index. The semantic search index is part of the Smart Search Tools framework and is implemented in the `SemanticSearchIndex` class. It includes methods to add files or URLs (web pages) to the semantic index and also allows the addition of fragments of plain text to the semantic index. The content from a tweet or an RSS feed item is added to the semantic search index when a new tweet or RSS item is downloaded from the Internet and is parsed for occurrences of concepts from the included domain ontologies. The links between concepts and the tweets/RSS-items in which they occur are added to triple store. The manager classes take care of adding the metadata describing the tweet or RSS item to the triple store.

The `SemanticSearchIndex` instance also provides methods for adding ontologies containing concepts with labels to the triple store. The labels are used to search for occurrences in the downloaded tweets or RSS-items.

To search the semantic search index a `search` method is implemented in the `SemanticSearchIndex` class. This method returns the results of a search query. The results may contain all of the different kinds of (PDF)documents, tweets, or RSS-items that were added to the index.

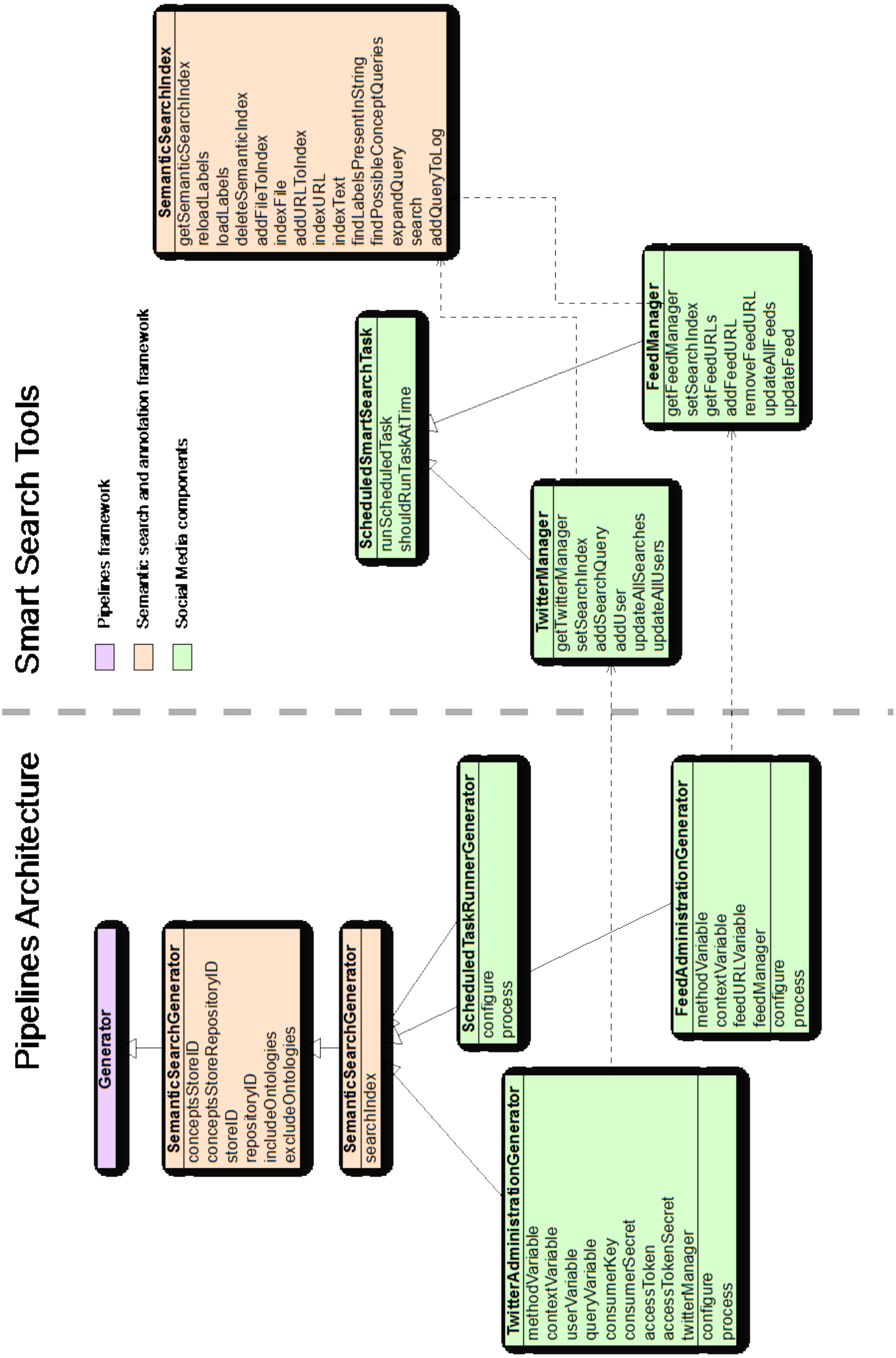


Figure 1The Class structure of the Social Media API

4.1 Pipeline implementation

The manager classes presented in the previous section can be used in different types such as web-applications or desktop applications. In the Information Management group at Food & Biobased Research we have developed a framework for fast and efficient web application development based on interconnected components. These reusable components are sequentially ordered in pipelines [12].

To be able to easily add the Social Media components to a web service implemented using the pipelines architecture, a set of components was created. These components can be configured in the pipelines configuration file as can other components that are part of the pipelines framework. Much of the functionality that is needed to present the Twitter/RSS results of a search query is already implemented in the pipeline components that were created as part of the Smart Search Tools and the generic pipeline components. Only component, generators, for administrative functions for Social Media needed to be added the framework. Instances of the `TwitterAdministrationGenerator` class use a reference to a `TwitterManager` instance to be able to add users and or Twitter-queries to the smart search tools. Likewise, instances of `FeedAdministrationGenerator` use an instance of `FeedManager` to be able to add RSS feeds. Within the configuration of `TwitterAdministrationGenerator`, the different security tokens and secrets can be added to be able to access Twitter.

4.1.1 Pipeline configuration file

Configuration of web services that add the administration functionality for the Social Media components is defined in the `pipelines.xml` file, for more information see [12]. To be able to use the Social Media components, we first need to define the processors in the preamble:

```
<generators>
     <generator type="feed-admin"
   class="nl.wur.fbr.smartsearch.pipelines.model.generators.FeedAdministrationGenerator"/>
     <generator type="twitter-admin"
class="nl.wur.fbr.smartsearch.pipelines.model.generators.TwitterAdministrationGenerator"/>
     <generator type="tasks"
   class="nl.wur.fbr.smartsearch.pipelines.model.generators.ScheduledTaskRunnerGenerator"/>
</generators>
```

Next we can define the pipelines. The options for the Social Media Pipeline components are:

Processor	Attribute/Element	Type	Description
All	`@store-id`	ID	The ID that identifies the triple store. The triple store is configured independently
	`@repository-id`	ID	The name of the repository in the triple store.
	`@index-namespace`	URI	The URI for the namespace/named graph to which the data will be added in

			the repository.
	`concept-repository/@store-id`	ID	The ID that identifies the triple store from which the domain ontologies used in annotation and indexing are loaded.
	`concept-repository/@repository-id`	ID	The name of the repository from which the domain ontologies used in annotation and indexing are loaded.
`TwitterAdministrationGenerator`	`@method`	name	add-user: To add users, add-query: To add a query to be put to twitter, update: run manual update of twitter feeds.
	`@query`	String	The query to be added.
	`@user-id`	String	The twitter id of the user to be added.
	`@consumer-key`	String	The consumer key provided by Twitter.
	`@consumer-secret`	String	The consumer secret provided by Twitter.
	`@access-token`	String	The access token provided by Twitter.
	`@access-token-secret`	String	The access token sectet provided by Twitter.
	`@twitter-context`	URI	The URI for the namespace/named graph to which Twitter data will be added in the repository.
`FeedAdministrationGenerator`	`@method`	name	add-feed: To add a feed (@feed-url) to the database, get-feed-list: Returns the list of feeds (URLs) in the database, update-feed: Updates the feed identified in @feed-url, update-all: updates all feeds, and force-update-all: updates all feeds irrespective of the update policy defined by the feed.
	`@feed-url`	URL	The URL of the feed.
	`@feed-context`	URI	The URI for the namespace/named graph to which Feed data will be added in the repository.

An example configuration file:

```
<pipelines>
    <pipeline>
        <name>Scheduled tasks</name>
        <description></description>
        <matcher>
            <matcher type="url" path="/{application}/run-tasks"/>
        </matcher>
        <generator type="tasks" store-id="smart-search-demo"
                   repository-id="smartsearch-index"
                   index-namespace="http://www.foodvoc.org/semantic-search-index/">
            <concept-repository store-id="smart-search-demo" repository-id="smartsearch"/>
        </generator>
        <transformer type="json"/>
        <serializer type="json"/>
    </pipeline>
```

```
     <pipeline>
         <name>Feed admin service</name>
         <description></description>
         <matcher>
             <matcher type="url" path="/{application}/feeds/{method}"/>
         </matcher>
         <generator type="feed-admin" method="$method"
                    feed-url="$query_url" feed-context="http://www.foodvoc.org/feeds/"
                    store-id="smart-search-demo" repository-id="smartsearch-index"
                    index-namespace="http://www.foodvoc.org/semantic-search-index/">
             <concept-repository store-id="smart-search-demo" repository-id="smartsearch"/>
         </generator>
         <transformer type="json"/>
         <serializer type="json"/>
     </pipeline>

     <pipeline>
         <name>Twitter admin service</name>
         <description></description>
         <matcher>
             <matcher type="url" path="/{application}/twitter/{method}"/>
         </matcher>
         <generator type="twitter-admin" method="$method" query="$query_query"
                     user-id="$query_user-id" consumer-key="XXXXXXXXXXXXXXXXXXXXXXX"
                     consumer-secret="XXXXXXXXXXXXXXXXXXXXXXX"
                     access-token="XXXXXXXXXXXXXXXXXXXXXXX"
                     access-token-secret="XXXXXXXXXXXXXXXXXXXXXXX"
                     twitter-context="http://www.foodvoc.org/twitter/"
                     store-id="smart-search-demo" repository-id="smartsearch-index"
                     index-namespace="http://www.foodvoc.org/semantic-search-index/">
             <concept-repository store-id="smart-search-demo" repository-id="smartsearch"/>
         </generator>
         <transformer type="json"/>
         <serializer type="json"/>
     </pipeline>
</pipelines>
```

5 Future Work

Although we believe that the two most important social media platforms for our purposes and target domains (RSS and Twitter) are now supported, other Social Media platforms such as Facebook, LinkedIn and Google+ will need to be added.

Other functionality that will need to be added is not specific to the Social Media components but is more general in that they are part of the semantic search framework in general. One may think of improved automatic annotation functionality through the extensive use of natural language processing techniques and better extraction of metadata.

6 References

1. Social media - Definition and More from the Free Merriam-Webster Dictionary. *merriam-webster.com* at <http://www.merriam-webster.com/dictionary/social%20media>
2. Boyd, D. M. & Ellison, N. B. Social Network Sites: Definition, History, and Scholarship. *JCMC* **13,** 210–230 (2007).
3. RSS. *en.wikipedia.org* at http://en.wikipedia.org/wiki/RSS
4. White, C. M. *Social Media, Crisis Communication, and Emergency Management.* (CRC Press, 2011).
5. Breslin, J. G., Passant, A. & Decker, S. *The Social Semantic Web.* (Springer Berlin Heidelberg, 2009). doi:10.1007/978-3-642-01172-6
6. Cherry Picker | Infographic: trust in blogs and social media. *cherrypicker.nl* at http://cherrypicker.nl/en/pr-marketing-eng/infographic-trust-blogs-social-media
7. Dublin Core Metadata Element Set, Version 1.1. *dublincore.org* (2012). at http://dublincore.org/documents/dces
8. Brickley, D. & Miller, L. FOAF Vocabulary Specification 0.99. *xmlns.com* (2014). at http://xmlns.com/foaf/spec
9. D'Arcus, B. & Giasson, F. **Bibliographic Ontology Specification**. *bibliontology.com* (2009). at http://bibliontology.com/specification
10. Burrueta, D. *et al.* SIOC ontology. *sioc-project.org* (2010). at http://www.sioc-project.org/ontology
11. **RDF Site Summary (RSS) 1.0**. *web.resource.org* (2000). at http://web.resource.org/rss/1.0
12. Willems, D. J. M. *Web Service Pipelines.* (2014), eFoodLab, COMMIT/ deliverable, July 2014.