Acta Universitatis Apulensis,
ISSN 1582-5329, 2011,
Special Issue on Understanding Complex Systems,
Eds. Barna Iantovics, Ladislav Hluchý and Roumen Kountchev

NEW DEVELOPMENTS ON THE NEWISTIC PLATFORM

Horatiu Mocian and Ovidiu Dan

Short paper

Abstract. Newistic is an aggregator platform that gathers news articles and blog entries from thousands of on-line sources. In addition to the extraction of text from HTML pages, articles are classified into predefined topics, and clustered. Also, named entities are extracted. This paper presents the new features, as well as improvements, that were added to the platform since the last publication detailing the system. These include support for multiple languages, multilabel categorization, a new HTML extraction engine, and geographical location disambiguation.

Keywords: *news aggregation, text mining, distributed systems, machine learning*

2000 *Mathematics Subject Classification*: 68M11, 68M14

1.Introduction

The goal of the Newistic project is to index and analyze all the freshly produced content, coming from any type of source: news websites, blogs, Twitter, and others. One of the big differences between indexing new content and indexing web content in general is the real-time aspect of the former: news should be gathered as quickly as possible after they are published, ideally in real-time. This is the reason why Newistic endeavours to crawl each new article in less than 10 minutes after it was published. However, to fulfil this objective, the system must be highly-scalable and distributed.

Newistic performs all the basic tasks of a news aggregator: crawling, text and image extraction, categorization and clustering. Each of these tasks is implemented by a different module that can be run independently, as a black box. The modular character of the architecture makes it very easy to distribute. All components except one (clustering) can be run with multiple instances, on different computers. Thus, the system can be scaled out to accommodate a large number of sources. We consider this one of the biggest strengths of the platform.

After developing the basic components of the news aggregator, we looked into expanding the feature set to more complex tasks. One of the first things was to add support for multiple languages, which raised two problems. First, a series of tools for parsing and stemming documents in different languages were required. For this, a wrapper class was created that uses the appropriate tools, as required by a given language. Second, different training sets needed to be used for categorization, and different index files needed to be maintained for each language. In summary, a new layer of information management was added. The other improvements and additions had a smaller impact on the overall design of the platform, but have associated issues nonetheless. They will be detailed in section 3.

The contributions of this paper are: an update to the modular and distributed architecture of Newistic, to make it more scalable, the description of a news extraction algorithm for HTML pages, as well as a geographical disambiguation algorithm.

The rest of the paper is organized as follows: Section 2 gives a review of the architecture of Newistic, while Section 3 describes all the new features in detail. Section 4 presents on-going and future work, while the paper concludes with Section 5.

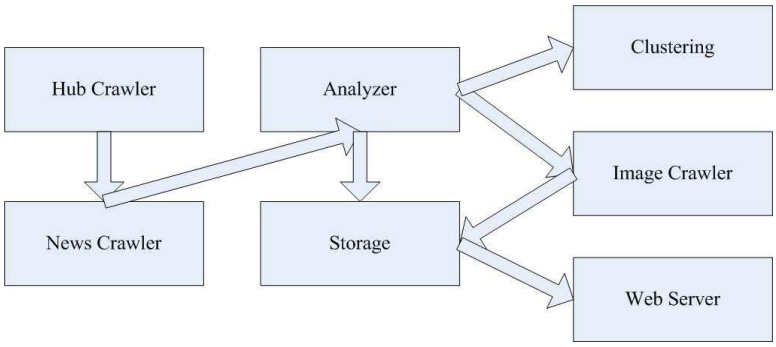

Figure 1: Architecture of the Newistic platform

2.Review of Architecture

This section provides a brief review of the architecture of Newistic, as it was described in [2]. The layout of the system can be seen in figure 1. We regard this architecture as a processing pipeline.

The first modules from the pipeline are the crawlers: HubCrawler, NewsCrawler and ImageCrawler. The HubCrawler scans each hub periodically for new links, that are sent to the NewsCrawler, which downloads the web page and sends it further down the pipe. Finally, the image crawler downloads all the text-related images that are found during HTML extraction and stores them in a local file structure.

The Analyzer module performs most of the text analysis. It starts by detecting the charset of each web page. Next, it tries to determine if the page contains a news article. This was initially done by an heuristic algorithm, which compared the amount text in the page to the quantity of other elements (menus, advertisements, etc). A new algorithm for extracting news articles will be described in Section 3. If the page contains an article, language detection is applied, and if it belongs to one of the accepted languages, categorization and named entity recognition are performed. For categorization, the kNN algorithm [5] was used, which proved to be better than the majority of traditional algorithms, but not as good as the computationally-intensive SVM algorithm [7]. Finally, named entity recognition was done using ANNIE from the GATE platform [1].

The Storage component keeps an index of the crawled news, so that they can be retrieved easily. The index was implemented using Lucene, an open-

source indexing software [3]. It supports queries with complex clauses, somewhat similar to Google search engine.

The Clustering component performs clustering at a fixed interval (around 5 minutes), covering the news gathered in the last 24 hours. The clusters are sent to the webserver for display, while duplicates detected from the same source are sent to the Storage component to be deleted. Duplicates from different sources are marked. For this task, QT clustering has been used, a high-quality algorithm that was borrowed from bioinformatics [4].

Finally, the Webserver component consists of a simple frontend that allows users to search news and view them by cluster and category. Now, this component has been replaced by an API, which can be used to build a range of applications (see Section 3).

3.Improvements and Additions to the Platform

This section describes the most important changes that were made to the Newistic platform in 2009. The new features are: support for multiple languages, implementation of an API, using a distributed index, performing geographical name disambiguation and developing a new extraction engine.

Support for multiple languages

One of the most important additions to the platform is the support for multiple languages. This required changes in the Analyzer, Categorization, Clustering and Storage/Indexing modules. More specific, each component has to deal with each language separately, so it has to keep a series of structures for it. For example, the clustering component builds the clusters for each language, while texts in different languages are needed for training the categorization algorithm. Moreover, the storage component keeps a separate index for each language. Currently, 5 languages are supported: English, Romanian, French, Greek and Serbian, while a few more are planned. For each of these languages, a stopwords lists, stemmer and transliterator were implemented. An aspect worth mentioning is the use of two alphabets by the Serbian language: Latin and Cyrilic. In this case, a transliterator was used to transform any text to one alphabet (Latin).

Application Programming Interface

The architecture described in the previous section provides a Webserver component that accesses the index directly and allows the user to perform different

searches. A layer of abstraction was added to this connection, in form of an API. After implementing the API, Newistic can be viewed as a web service: it can be accessed by an external application over the web using the universal XML-RPC format. The calls that can be made through the API include complex search queries, getting the list of sources used by the system or calculating counts for certain fields in a query. Being based on the XML-RPC format, the API can be accessed by any popular web programming language/platform, like ASP.NET, PHP, Ruby and Python. It can be used as a building block for more complex portals, or for deeper text analysis applied on the articles crawled by Newistic. A real-life example built on top of Newistic API can be seen at `http://prezidentiale.cotidianul.ro`, which shows news about the candidates to the Romanian presidential election in 2009.

Distributed index

Another addition to the platform is constituted by implementing a distributed index. For this, we have used Apache Solr, an enterprise search tool built on top of Lucene, that allows replication and sharding, among others. Thus, the index can now be distributed on multiple servers in a transparent manner. Also, we have introduced the concepts of active server and historic server. The active server stores the news from the last 30 days (one month), while the historic servers store news for a longer period of time.

Geographical names disambiguation

This module implements two functionalities:

1. For a list of strings that may or may not be geographical locations, determine if each element of the string is a geographical location. For each determined location, return its unique numerical Geonames ID [1] and its hierarchy (for example, for Easton, a possible hierarchy would be Globe > North America > USA > Pennsylvania > Easton).

2. If a geographical location is ambiguous (there are several towns named Easton in the US), it tries to disambiguate it (it tries to determine which of the possible locations with the same name is the one mentioned in the list). The disambiguation is based on several factors. First, if locations belonging to a certain country or division are found in the text, these are

[1] http://www.geonames.org

considered clues for ambiguous locations. Second, the country or region of the website that is crawled can be considered as a clue. Third, if there is a need to choose between two or more possible locations, without any other clues, the largest of them has the biggest chance of being the correct one.

A different approach to geographical names disambiguation, based on Wikipedia, can be found in [6].

New extraction engine

We have implemented a more accurate extraction engine, that uses XPath. It requires some manual training for each source (which takes around 10 minutes to perform). In summary, it needs to know what HTML elements (e.g. div, h1, img) contain the title, text and images from a news article. They either have to have a specific ID, or belong to a certain class. Using these elements, rules are defined for extracting the title, body and images. Once these rules are obtained, an XPath processor parses each page recursively and retrieves the information found in the elements defined by the respective rules. This method is easy to train and highly accurate.

Current architecture of Newistic

In addition to the improvements that were presented in this section, several changes were made to the architecture. First, the Categorization module was decoupled from the Analyzer component, and is now a stand-alone component that can be accessed through XML-RPC. The Geographical Disambiguation module is also independent and is accessed through XML-RPC. The Storage component is now called Indexer, and communicates internally with the Solr servers. Thus, the Indexer component is only a wrapper of the real servers. Finally, the Webserver component becomes an external component that communicates with the API, in the same way as any other application that is built over this system. Thus, it is not considered as a component of Newistic anymore.

4. Work In Progress and Future Work

We are currently working on a parallel/distributed implementation of QT clustering, which, if successful, would make the Newistic platform fully distributed. We aim to obtain linear scaleup with our parallel implementation

and we have almost achieved it. We have tried several ways of parallelizing the algorithm, and the results will soon be submitted to a conference.

Another functionality that will be added soon is to crawl feeds from Twitter. Although crawling the entire output from Twitter is a big challenge, and needs a dedicated system built for it, crawling only feeds of popular users and channels is perfectly achievable on the current platform, and a necessary addition if we follow the objective of crawling any source of fresh information, as Twitter is establishing itself as a major platform for delivering real-time awareness.

A third addition that is necessary to the Newistic platform is sentiment analysis. In its most simple form, this task has to establish whether a certain article is positive or negative. In more complex scenarios, this task establishes other sentiments, too (fear, anxiety). We believe that this functionality is becoming increasingly necessary for any text mining platform, and it becomes even more useful when applied on news articles.

5.Conclusion

This paper has reviewed the core functionality of Newistic, and it has described new developments on the platform. Our goal is to make this platform as complete as possible, and transform it into a powerful tool for analyzing real-time content. We are continuously improving the platform, and will continue to add even more useful features.

Currently, the Newistic platform is running on three machines, gathering news from approximately 1500 news sources in five languages. It has been running without major disruptions for more than 6 months.

References

[1] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

[2] Ovidiu Dan and Horatiu Mocian. Scalable web mining with newistic. In *Proceedings of PAKDD '09*, 2009.

[3] Erik Hatcher and Otis Gospodnetic. *Lucene in Action (In Action series)*. Manning Publications, December 2004.

[4] L. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: Identification and analysis of coexpressed genes. *Genome Research 9, pp. 1106-1115*, 1999.

[5] Brij Masand, Gordon Linoff, and David Waltz. Classifying news stories using memory based reasoning. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 59–65, New York, NY, USA, 1992. ACM.

[6] Simon Overell and Stefan Rüger. Using co-occurrence models for placename disambiguation. *Int. J. Geogr. Inf. Sci.*, 22(3):265–287, 2008.

[7] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49, New York, NY, USA, 1999. ACM.

Horatiu Mocian
Department of Computing
Imperial College London
e-mail:*horatiumocian@gmail.com*

Ovidiu Dan
WUME Laboratory
Lehigh University, PA, United States
e-mail:*ovidiu.dan@lehigh.edu*