Finding news in a haystack — Event based news clustering with social media based ranking

Martin Weber and Maarten H. Lamers, martin.weber@gmail.com, lamers@liacs.nl
Media Technology MSc program,
Leiden Institute of Advanced Computer Science,The Netherlands

Abstract— **We present NewsClu, an approach to collect, cluster, categorize and select news articles from the Internet. The framework is not limited to a certain field, resort or division of news – it is capable to work on any topic and runs on real time live data on the Internet. NewsClu keeps its user *informed* about that particular topic by filtering, organizing, ranking and clustering of news according to static, dynamic and social factors. Through event based clustering the framework filters duplicate and less relevant news, making it easier to read more interesting news in less time. NewsClu is innovative in terms of clustering and ranking compared to other approaches.**

Keywords: ***NewsClu, News, Aggregation, Clustering, Text Similarity***

I. Introduction and Problem Statement

In the pre-Internet-era, news selection, organization and filtering were within the main responsibilities of a journalist (Gans (1979)). In recent years, the Internet has become a mainstream medium (Morris & Ogan (1996)) and offers possibilities to easily produce and distribute content. Several paradigms in news distribution shifted since then: The coverage of news on the Internet is no longer limited to a certain amount of pages or time and the recipient can get diverse coverage of news from various sources easily and fast. Also, syndication of content has become easier and faster while the selection of news is (partially) done automatically by algorithms. But with more news and sources than ever, it has not become easier to find relevant news on a specific topic.

In Table 1 we classify different approaches of news broadcastings by their scope. If one is interested in news on a specific topic like »North Korea«, »iPhone«, »François Hollande« or »Adele«, several techniques to gather news from different scopes can be applied. Regardless which approach is taken and which services are used, one may be confronted with multiple news about the same event within the field of interest.

Two main factors contribute to those duplicates: firstly, the number of sources one fetches the news from. The bigger the number, the higher the chance of receiving duplicate stories covering the same event. Secondly, the importance of events within the field of specialized or general news. At some point of attention, high peaks of interest will occur. Indicative for those high peaks is that coverage about special interest topics breaks through to mainstream-media, i.e. (*World News* or *Resort News*). Thus, coverage at the peak-point of interest about an event in the field is significantly higher than in times with lower interest. McCombs & Shaw (1972) describe this phenomenon in mass media as *Agenda Setting*: After the peak, the topic-specific news will gradually decline and after some time not be covered in the mass media anymore.

Especially when such a peak brings extended and multiplied coverage, one is confronted with manually filtering, selecting and reviewing those news. The fact that publishers or newspapers often build stories (partially) on texts from news agencies aggravates this situation: Through the pure mass of news around the peak one may easily fail in finding *good* and *relevant* news about the event besides the top news that is covered all over. In the context of this work, we view *relevance* as follows: »something $\mathcal{A}$ is relevant to a task $\mathcal{T}$ if it increases the likelihood of accomplishing the goal $\mathcal{G}$, which is implied by $\mathcal{T}$«. (Hjørland & Christensen (2002)) In terms of news retrieval, $\mathcal{A}$ is an unread news article covering an event in the field of interest $\mathcal{T}$. The goal $\mathcal{G}$ can be described as the need to be informed for professional, personal or other reasons.

As of today, there is no sophisticated automated method to filter, cluster and rank relevant news regarding a specified topic on a timely basis. We aim to solve this problem through the development of our algorithm.

II. Existing Approaches

We want to briefly introduce the most important frameworks for news discovery, filtering and aggregation and discuss the approach they take.

978-0-7695-5114-2/13 $26.00 © 2013 IEEE
DOI 10.1109/CGC.2013.57

Table 1. Classification of news through various media

Scope	Approach, Specifity	Examples
Global	Limited amount of news according to users region & timeframe, Newsblaster, Google News, Yahoo News, Bing News	Newspapers, magazines, news web sites, news broadcast on TV/radio
Resort	Politics / Business / Local / Sports / Feuilleton / Technology, Entertainment / Music / Leisure. . .	(Sub)sections of web sites, Weblogs, Dedicated web sites, Rivva, Blogrunner, Sections of Newsblaster / Google News
Sub-Resort	Specialized areas / topics within resorts e.g. Sports ⇝ Soccer, Technology ⇝ Apple	Specialized Sites (editorial) e.g. football365.com, 9to5mac.com
Social	Most liked / linked / favorited stories in timeframe	Twitter, favstar.fm, Google+, Rivva, zite, trap.it, Fever, Wavii
Social (personal)	Linked / liked / favorited stories from social peers (and evt. outsiders) in social networks	Google News (personalized version) Instapaper, zite, trap.it, Facebook, Twitter
Keyword-based	Filter streams of news in timeframe according to fixed keyword(s) / interest(s) or rules	Yahoo Pipes, Google Alerts, Google API, advanced / custom Google Search
Mashups	mix of two or more of the above mentioned techniques to filter, aggregate, cluster or rank news	Yahoo Pipes, trap.it, Zite, Rivva, Techmeme Newsblaster (content summaries)

Columbia Newsblaster[1] was the predecessor to Google News. It gives a fully algorithmically overview about events of the last day, but seems partially abandoned since several features do not work correct anymore. It crawls, clusters and sorts news from 17 sources into six main categories with methods of topic detection and NLP (McKeown et al. (2002)). A summary for each story is generated automatically.

Google News[2] aggregates news from more than 25,000 sources (Cohen (2009)) and groups content automatically in sections. The front page shows the top stories of the day according to the user's geographical region, language and settings. The *full coverage* of a story, resulting in dozens to thousand of similar stories can be shown. While selection and story or event-based clustering works well for content featured on the front page, it doesn't for individual topics[3]. This makes it hard to find diverse coverage on custom topics.

Google Alerts[4] monitors the web for new content on a specific keyword / topic and sends the user an email with links to the results. Alerts is potentially slow, since the fastest notification interval is once a day. Furthermore, it may not be reliable for topics with a high volume of news since the maximum number of results covered in each email is limited.

Yahoo! Pipes[5] is a tool to aggregate, manipulate, and mashup content. A *pipe* can be created by combining content sources, filters and events. Yu et al. (2008) describe Pipes as a valuable and unique tool inside the Internet's mashup-world. Pipes does not have a user interface for results; the output can be accessed via RSS- or JSON-feeds.

Zite, **trap.it** and **Wavii**[6] are personalized aggregators for news on regions of interest (Zite, Wavii) or specified keyword / topics (trap.it). With Zite and Wavii, the user can claim sections of interest, such as *World News* or *Entrepreneurship*, but not individual topics or stories; see Zite-Team (2012). Thus, general topics like *Syria* show up, but more special ones like *François Hollande* don't. Trap.it is focussed on the user's individual topics of interest. Users set *traps* and the service *catches* news on these topics. Trap.it lacks a good duplicate detection and isn't capable of clustering stories together.

Fever[7] crawls a user-provided list of news-feeds and calculates the *temperature* of single items – determined by the number of links an item has on other stories within the sources. Thus, the ranking is a closed-world-system, specially made for the user's preferences and sources. This calculation works as long there is *one* original source to an event. If the event is not covered on a single origin web page, the system fails, since it is not capable of clustering similar news to one event. The software also does not incorporate social network data.

Rivva and **Techmeme**[8] are news-aggregators for trending topics frequently linked in social networks. Farber (2007) describes Techmeme as »a one-page, aggregated, filtered, archiveable summary in near real-time of what is new and generating conversation«. The services work through combination of both algoritmic and human editing to generate featured stories. Their general functionality works through scraping trusted, parsable web-content and displaying items with the backlinks and / or social mentions. Both services create clusters of stories around events and rank one of the corresponding items as the clusters head. The difference between Fever and Rivva / Techmeme is the general domain of those two web services: The crawlers are open to any source[9]. Both Rivva and Techmeme do not work in terms of news-retrieval on specific keywords.

[1] `http://newsblaster.cs.columbia.edu`

[2] `http://news.google.com`

[3] *All sources* often covers different events and ties them together.

[4] `http://google.com/alerts`

[5] `http://pipes.yahoo.com`

[6] See `http://zite.com`, `http://trap.it` and `http://wavii.com`.

[7] `http://feedafever.com`

[8] `http://rivva.de` and `http://www.techmeme.com`

[9] *Rivva* originally started with only seven source weblogs and expanded its network *Web Of Trust* in those sources. (F. Westphal, personal communication, February 21, 2012).

III. NewsClu in Detail

NewsClu builds upon well-known techniques and algorithms to achieve its goal of keeping the user informed about a certain topic. The software consists of modular components for crawling, collecting, filtering, clustering, ranking and selecting news items that match the desired topic. The software is capable of processing content in English language which is available in a textual form.

A. Crawling of URLs

Because it is virtually impossible to crawl sufficient amounts of content from the web with limited bandwidth and hardware, NewsClu relies on a selection of companies providing and aggregating news on the web: Google News, Bing News, Yahoo News[10]. We also use Google's Advanced Search, Google Blogsearch and Icerocket[11] to crawl content from parts of the web that might not be covered by the aforementioned services.

B. Crawling the Content

A `cURL` based self-written web-crawler processes the list of to be crawled web pages and extracts the textual main content of the web page using a PHP-port (Minoukadeh (2010)) of *Readability*[12]. This software converts a rich, full web page to its main textual content. All links within the content are also processed and queued for crawling and marked with a `depth` of 1, so links on those pages will not be queued again. Then, the text-tag-ratio Q_{TTR} of the main textual content is calculated:

$$Q_{TTR} = \frac{|P|}{|F|} \tag{1}$$

Where P is the length of the processed plain text and F is the pages full length, containing HTML-markup. The Q_{TTR} with a range from 0—1 defines the fraction of plain text within the news' corpus. We assume that news tend to be more valuable if they are not cluttered with a high amount of markup, since we made this observation during several test runs: pages with lower Q_{TTR} were intersected with other elements (e.g. image galleries, asides with links to other pages,...), possibly to generate more clicks. While crawling and processing news, NewsClu uses a number of helper functions to ensure the quality of the content is feasible. The helper functions consist of a list of blacklisted domains, text-wise duplicate detectors, limitation of the content-length and keyword and host-based blocking of URLs. Hereafter we refer to the collection of all crawled web pages as D, where d is a single document in this collection.

[10] See `https://news.google.com`, `http://www.bing.com/?scope=news` and `http://news.yahoo.com`.

[11] See `https://developers.google.com` (Limited access to Google's Advanced Search), `https://www.google.com/blogsearch` and `http://icerocket.com`.

[12] `http://readability.com`

C. Similarity of Items

To cluster news, a similarity measurement for all $d \in D$ must be generated (see Segaran (2007), Metzler et al. (2007), Corley & Mihalcea (2005) and Pedersen et al. (2004)). Since approaches like `Levenshtein-distance` with $O(n \cdot m)$ from Levenshtein (1966) or `similar_text` with $O(n^3)$ (n = length of the longest chunk) from Oliver (1993) are limited and costly, we do not use them. Metzler et al. (2007) showed that »a simple hybrid technique that combines lexical, stemmed, and probabilistic matches results in far superior performance than any method alone.« We follow this approach with our similarity measurement for each pair $(d_i, d_j) \in D$ in the database younger than 48 hours (so the number of computations is $|D|^2$). As described in Andrews & Fox (2007) and Metzler et al. (2007), several preprocessing-steps make the processing more meaningful. They consist of:

- **Filtering** Removal of markup, punctation and special characters from sentences.
- **Tokenization** Splitting of text into individual chunks. We treat each word as a chunk, using a *bag of words* approach.
- **Stemming** Reduction of words to their base form using the algorithm from Porter (1980).
- **Stop word removal** Deletion of chunks that do not convey meaning. We used the stop word list by Oracle (2012), since it is widely used and freely available.
- **Pruning** Removal of words that appear with a low frequency throughout the text. Instead of a fixed pruning ratio, we prune words that occur only *once* in the text. The topic is also pruned, since it does not add meaningful content to the word list. This sliding-window-approach is more robust for texts with a big variety in length: If we would have chosen a fixed pruning ratio (i.e. delete the X% of words with the lowest occurrences), this would lead to too small results on short texts and on too big results on longer texts.

The result of these pre-processing steps is a set of tuples containing *golden words* and their frequency. For each document $d \in D$ we compute a term frequency vector f_i out of these tuples. Let A and B hold those term frequency vectors $f_i = f(d_i)$ and $f_j = f(d_j)$, where $f_{i(w)}$ defines the frequency of the word w in a document d_i. The complexity of one

calculation is $O(A \cdot B)$. The similarity between the tuples of the two texts' golden words is measured by calculating the cosine of the angle between A and B (Manning et al. (2008)), it ranges from 0 to 1, since the frequencies cannot be negative. Using the vector space model together with cosine similarity, we compute the similarity matrix M for the document database. The cosine similarity of A and B is defined (using the *Euclidean dot product formula*) as

$$similarity(A,B) = \frac{A \cdot B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad (2)$$

The resulting matrix M then consists of similarity values for all text-pairs.

D. Similarity Based Clustering

We cluster together all $d \in D$ with a high similarity value, since we presume them to deal with the same event. The choice of the similarity measure has an influence on clustering, see Strehl et al. (2000). Since NewsClu does not focus on a specific approach, domain or keyword, the similarity and clustering must be independent from those. Clustering algorithms such as `k-means` from Hartigan & Wong (1979) are not applicable because they have limitations[13]. Therefore, we use an adopted algorithm based on Agglomerative Single Link Clustering, as described in Jain et al. (1999). To cluster texts more fine-grained, we use two different stages that we briefly discuss here. Both stages have similarity thresholds, θ_1 and θ_2. Experiments have shown that $\theta_1 = 0.8$ and $\theta_2 = 0.6$ are good starting points.

Clustering, Stage One

1. For each pair (d_i, d_j) where $d_i, d_j \in D$ with similarity $\geq \theta_1$ do the clustering as follows:
2. Find a cluster ID *cid*:
   - If either $d_i.cid$ OR $d_j.cid$, use this *cid*.
   - If $d_i.cid = d_j.cid$ use $d_i.cid$.
   - If $d_i.cid \neq d_j.cid$, use smaller cid_{low}. $\forall\, d \in D$ with cid = cid_{high} assign cid_{low}.
   - If $d_i.cid$ AND $d_j.cid = 0$, get a new *cid* by incrementing the $max(cid)$ by 1.
3. Assign *cid* to d_i and d_j. Repeat.

Clustering, Stage Two

1. For each pair (d_i, d_j) with similarity $\geq \theta_2$ and $< \theta_1$ do the clustering as follows:
2. Check the criterion for appending: If either $d_i.cid$ OR $d_j.cid$, use this *cid*.
3. Assign *cid* to d_i and d_j. Repeat.

[13] With k-means, the number of clusters k has to be known *a priori*. Furthermore, k-means' pattern-detection (Andrews & Fox (2007); p.8) is not very advanced.

E. Link Based Clustering

In addition to similarity based clustering, we use link-based clustering by connecting all d based on the existence of hyperlinks. This allows the detection of *Link Love*[14] and clusters such documents together, even if they do not have a high word-wise similarity.

F. Ranking

The ranking for each d is based on a number of *fixed factors* and the *social inclination* of the d's mentions on social networks over time. Since the clusters are computed earlier, the ranking sorts the members of a cluster and all clusters themselves by comparing their highest ranking members (cluster heads).

Fixed Factors

We utilize several fixed factors for each d.

- **Twitter** No. of *mentions* and *retweets* d has.
- **Facebook** Sum of *likes*, *shares* and *clicks* d has.
- **Age**The documents publication date (if known); used as punishment-criterion. If the date is unknown, we use the crawling time[15] of the content.
- **Text-Tag-Ratio** The Q_{TTR} is used as a criterion for the text quality with a threshold of 0.6. Texts with lower Q_{TTR} receive a negative score.
- **Alexa** The Alexa ranking (Lo & Sedhain (2006)) of the content's domain is used as an estimate of how popular the web page is.

Social Inclination

We consider both Facebook and Twitter as important factors for the distribution and trendiness of news through the Internet. Thus, we measure the inclination of social mentions $S(t)$ on these services over time t, once each hour. After few hours, items reveal their trendiness. We consider a d to be more relevant if a high number of users on Facebook or Twitter recommend it in a short timespan. Looking only at momentary values, this could not be revealed. The inclination over timespan $t_1 - t_2$, for both Facebook and Twitter, can be calculated with the following formula:

$$\mathcal{I} = \frac{M(t_2) - M(t_1)}{t_2 - t_1} \quad (3)$$

The *fixed factors* and the *social inclination* $\mathcal{I}$ are combined into a ranking. We determined the weights of the single factors experimentally.

[14] *Link Love* is the description for a trending event that has its source on site d_i and spreads virally. Other sites now place links to the source site ($d_i \leftarrow d_j$). Regardless of the text in d_j, the connection is recognized.

[15] A d without publication date can't be cluster head or top news of the day, because we don't want to show possibly old news as new.

IV. Challenges of the Internet as Environment

It is of utter importance to realize that NewsClu runs *in the wild* on the Internet. This is fundamentally different from runs with a predefined (and somehow predictable) set of test-data such as *Reuters-21578* test collection database for text categorization research from Lewis (1997). Furthermore, all news sources can be completely unknown a priori. This holds for the content's structure, type and markup[16]. The complete absence of this information combined with the Internet's heterogeneity and decentralization requires robustness and tolerance against errors for NewsClu. However, each and every robustness rule can lead to possibly omitted content, thus it shall be clear that there is no such thing as *the list of perfect results* in a way that it could not be improved further. The age of the news is an important factor for meaningful results. As of today, there is no standardized method of accessing the publication date of a text published on a web page. We wrote a program to *guesstimate* the date of publication from the `Document Object Model`. The problem stays, however, partially unsolved.

V. Experiments and Evaluation

Because NewsClu is capable of running on any topic, some parts of the software need to be adjustable by the user (or an expert) to ensure best results. In general, broader topics with more news need a lower θ_1 (leads to less clusters) than specific topics (higher θ_1, leads to more clusters). The different weights of the ranking-factors in conjunction with the users preferences also play a role here.

We let people test NewsClu on their favorite topics[17] with live data on the Internet as of the time of testing (2012-05-22–2012-06-08; one test after each other). The evaluation was informal due to the low number of participants (n=13) and the qualitative approach. We suggested to change their topic if we found it to be ineligible. This is the case if the amount of news is too small to build meaningful clusters or if the topic is too general (i.e. »restaurant«), with local differences (i.e. »weather«) or with a non-unique meaning (i.e. »Ironman«). For each topic and test, we adjusted the similarity thresholds θ_1 and θ_2 experimentally according to a plausibility check whether the clustering seemed to be the most meaningful and seconded this by scrutinizing the topic's similarity histogram. We did not change the weights of the fixed factors during the evaluation.

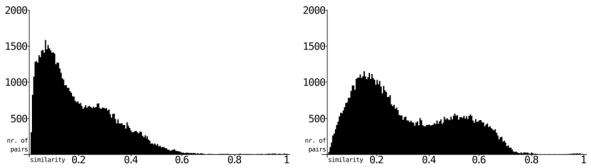
[16] For example, *Newsblaster* has a fixed list of sources which allows rule-based parsing and generation of abstracts according to the structure of the corresponding source.

[17] These topics were *Japan*, *iPhone* (2 test persons), *"Chen Guangcheng"*, *"Euro 2012"* (2 test persons), *"Facebook IPO"*, *Facebook Privacy*, *Flame*, *Fukushima*, *iOS*, *"iPhone 5"*, and *"spain protest"*.

The evaluation focussed on four questions. **Firstly**, if the desired topic could be mapped *as complete as possible* within NewsClu. This issue was evaluated by letting the testers compare the output the software against another news aggregators or the strategy of news retrieval of their choice, e.g. Google News or their personal news feeds. **Secondly**, if the *inter-cluster-difference* was reasonably good. This defined the expression of topics per cluster. Thus, no topic should be addressed within two clusters at once, each cluster should only represent *one* topic. **Thirdly**, if the *intra-cluster-coherence* was high enough to convince the test persons that members of one cluster only dealt with one topic and did not consist of stories dealing with different events. **Fourthly**, if the *ranking* between the news items and clusters was logical and reasonable.

Overall, the users were satisfied with the results of the software. They felt sufficiently informed about their topics and had not the feeling that important news were missing. Nearly all testers said they found relevant news through our framework that they would not have had found with their current method of news retrieval. Furthermore, the majority of users said that NewsClu kept them informed faster and / or more complete than their current method. The users also valued the lead news on top, saying that it covered the most important event within the topic.

Fig. 1. histograms of similarity pairs $(d_1, d_2) \in D$



(a) topic »*North Korea*« (b) topic »*Flame*«

The quality of the output[18], the intra-cluster coherence and inter-cluster-difference in particular varied during the test with different topics. In Fig. 1, histograms of NewsClu running on different topics are shown, with the similarity on the X-axis and the number of comparison pairs on the Y-axis. One can see the difference between well clusterable topics (Fig. 1a) and not so well clusterable topics (Fig. 1b)[19] by comparing the spread of values. Well clusterable

[18] Measured based upon the user's perception that compared NewsClu with the approach(es) he / she currently uses on the specific topic.

[19] Fig. 1b; topic *Flame*: The news retrieval was done just one day after the discovery of a new computer-virus entitled *Flame*. Thus, unique content or diverse articles weren't there then, most of the stories linked to one source. The two peaks are a result of *Flame*s multiple meanings, the *Olympic flame* and the computer virus.

topics have most of their pairs below the lowest clustering threshold, making it easy to find and cluster the *good* news. Not well clusterable topics resulted from news that were not diverse enough. This fuzziness was represented through one more more peaks with medium similarity, making clustering using this similarity values less meaningful and harder.

A. Output

NewsClu is a web page with the news of the last two days, the news of the day is the lead. Below, clusters of stories are presented. Each cluster represents a single event within the topic and has a head, which is the member of that cluster with the highest score. These heads ideally represent the clusters and their event in an adequate way and should be descriptive enough to summarize the clusters' content. The full list of members can also be revealed. The arrangement of clusters is based on the score of the cluster heads. The top news is the cluster head with the highest score of all.

VI. Discussion

We have shown that NewsClu is innovative and valuable in terms of sorting, filtering and clustering keyword-specific content from the web. The proposed approach is new and we consider NewsClu to be the first of its kind to offer these results. The software allows users to quickly gain an overview on a topic of their choice and events within the topic. Looking into the future, some components could be improved: The clustering-thresholds could be computed automatically through analyzing the specific similarity histograms (see Fig. 1) of the topic. In the current state, the manual (re)setting of those thresholds for each topic weakens the black-box approach of the framework a bit: Although it runs unsupervised and with live data, manual adjusting is needed there.

Furthermore, several settings could be improved: The now fixed ranking-factors could be set individually upon the user's preferences. NewsClu could be linked to the user's accounts on social media websites and could utilize data from friends of the user as additional ranking factors. Thus, news appearing those feeds could be ranked higher / differently. On input-level, it could be possible to add own sources or block unwanted ones. On output-level, it could be possible to vote clusters and / or their members up and down and to merge clusters. Through these user-interactions the system could profit from the user's expertise and could use this information to generate even more meaningful results.

References

Andrews, N. & Fox, E. (2007), 'Recent developments in document clustering', *Technical Report TR-07-35, Computer Science, Virginia Tech* pp. 1–25.

Cohen, J. (2009), 'Google News Blog: Same Protocol, More Options for News Publishers'.
URL: *http://googlenewsblog.blogspot.com/2009/12/same-protocol-more-options-for-news.html* (accessed 02/05/12)

Corley, C. & Mihalcea, R. (2005), Measuring the semantic similarity of texts, *in* 'Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment', number June, Association for Computational Linguistics, pp. 13–18.

Farber, D. (2007), 'Daylife's rather lifeless news aggregator'.
URL: *http://www.zdnet.com/blog/btl/daylifes-rather-lifeless-news-aggregator/4225* (accessed 13/02/12)

Gans, H. (1979), *Deciding what's news: a study of CBS evening news, NBC nightly news, Newsweek, and Time*, Northwestern University Pres.

Hartigan, J. & Wong, M. (1979), 'Algorithm AS 136: A k-means clustering algorithm', *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **28**, 100—-108.

Hjørland, B. & Christensen, F. S. (2002), 'Work tasks and socio-cognitive relevance: A specific example', *Journal of the American Society for Information Science and Technology* **53**(11), 960–965.

Jain, A., Murty, M. & Flynn, P. (1999), 'Data clustering: a review', *ACM computing surveys (CSUR)* **31**(3).

Levenshtein, V. (1966), 'Binary codes capable of correcting deletions, insertions, and reversals', *Soviet physics doklady* .

Lewis, D. D. (1997), 'Reuters-21578 text categorization test collection Distribution 1.0'.
URL: *http://www.daviddlewis.com/resources/testcollections/reuters21578/* (accessed 04/05/12)

Lo, B. & Sedhain, R. (2006), 'How reliable are website rankings? Implications for e-business advertising and internet search', *Issues in Information Systems* **7**(2), 233—-238.

Manning, C., Raghavan, P. & Schutze, H. (2008), *Introduction to information retrieval*, online edi edn, Cambridge University Press, Cambridge, England.

McCombs, M. E. & Shaw, D. L. (1972), 'The agenda-setting function of mass media', *The Public Opinion Quarterly* **36**(2), 176—-187.

McKeown, K. R., Barzilay, R., Evans, D., Hatzivassiloglou, V., Klavans, J. L., Nenkova, A., Sable, C., Schiffman, B. & Sigelman, S. (2002), 'Tracking and summarizing news on a daily basis with Columbia's Newsblaster', *Proceedings of the second international conference on Human Language Technology Research* pp. 280—-285.

Metzler, D., Dumais, S. & Meek, C. (2007), 'Similarity measures for short segments of text', *Advances in Information Retrieval* pp. 16—-27.

Minoukadeh, K. (2010), 'PHP Port of Arc90s Readability'.
URL: *http://www.keyvan.net/2010/08/php-readability/* (accessed 06/05/12)

Morris, M. & Ogan, C. (1996), 'The Internet as mass medium', *Journal of Computer-Mediated Communication* **1**(4).

Oliver, I. (1993), *Programming classics: implementing the world's best algorithms*, Prentice Hall.

Oracle (2012), 'MySQL 5.5 Reference Manual - Full-Text Stopwords'.
URL: *http://dev.mysql.com/doc/refman/5.5/en/fulltext-stopwords.html* (accessed 24/04/12)

Pedersen, T., Patwardhan, S. & Michelizzi, J. (2004), WordNet:: Similarity: measuring the relatedness of concepts, *in* 'Demonstration Papers at HLT-NAACL 2004', Association for Computational Linguistics, pp. 38–41.

Porter, M. (1980), 'An algorithm for suffix stripping', *Program: electronic library and information systems* **14**(3), 130—-137.

Segaran, T. (2007), *Programming collective intelligence: building smart web 2.0 applications*, Vol. 14, O'Reilly Media, Inc.

Strehl, A., Ghosh, J. & Mooney, R. (2000), Impact of similarity measures on web-page clustering, *in* 'Workshop on Artificial Intelligence for Web Search (AAAI 2000)', pp. 58—-64.

Yu, J., Benatallah, B., Casati, F. & Daniel, F. (2008), 'Understanding mashup development', *Internet Computing, IEEE* **12**(5), 44–52.

Zite-Team (2012), 'Zite - Blog: Zite Under the Hood'.
URL: *http://blog.zite.com/2012/01/zite-under-hood.html* (accessed 02/05/12)