Automating the Selection of Stories for *AI in the News*

Liang Dong[1], Reid G. Smith[2], Bruce G. Buchanan[3]

[1] School of Computing, Clemson University, SC, USA, ldong@clemson.edu
[2] Marathon Oil Corporation, rgsmith@marathonoil.com
[3]Computer Science Department, Univ. of Pittsburgh, PA, USA, buchanan@cs.pitt.edu

Abstract. It is relatively easy, albeit time-consuming, for a person to find and select news stories that meet subjective judgments of relevance and interest to a community. NewsFinder is an AI program that automates the steps involved in this task, from crawling the web to publishing the results. NewsFinder incorporates a learning program whose judgment of interestingness of stories can be trained by feedback from readers. Preliminary testing confirms the feasibility of automating the service to write *AI in the News* for the AAAI.

Keywords: News crawler, machine learning, supervised classification, SVM, artificial intelligence, AAAI, AITopics

1 Introduction

Selecting interesting news stories about AI, or any other topic, requires more than searching for individual terms. The AAAI started collecting current news stories about AI and making them available to interested readers several years ago, with manual selection and publishing by an intelligent webmaster.

Current news stories from credible sources that are considered relevant to AI and interesting to readers are presented every week in five different formats: (i) posting summarized news stories on the *AI in the News* page of the AITopics web site [2], (ii) sending periodic email messages to subscribers through the "AI Alerts" service, (iii) posting RSS feeds for stories associated with major AITopics, (iv) archiving each month's collection of stories for later reference, and (v) posting each news story into a separate page on the AITopics web site.[2]

Manually finding and posting stories that are likely to be interesting is time-consuming. Therefore, we have developed an AI program, NewsFinder, that collects news stories from selected sources, rates them with respect to a learned measure of goodness, and publishes them in the five formats mentioned. Off-the-shelf implementations of several existing techniques were integrated into a working system for the AAAI.

---

[2] Anyone may view current and archived stories and subscribe to any of the RSS feeds; Email alerts are available only to AAAI members.

Traditional recommender systems [9] require recording a user's preference and using techniques such as non-negative matrix factorization [12] to find users with similar tastes. Then, recommendations are based on the preferences of similar users. In our approach, we learn the characteristics of the items preferred by users and classify new items with respect to those.

The NewsFinder Program

The work of NewsFinder is implemented in four loosely-coupled program modules as in Fig. 1: (A) Crawling; (B) Training; (C) Ranking; (D) Publishing. The first three are independent from each other and the last two usually run together.

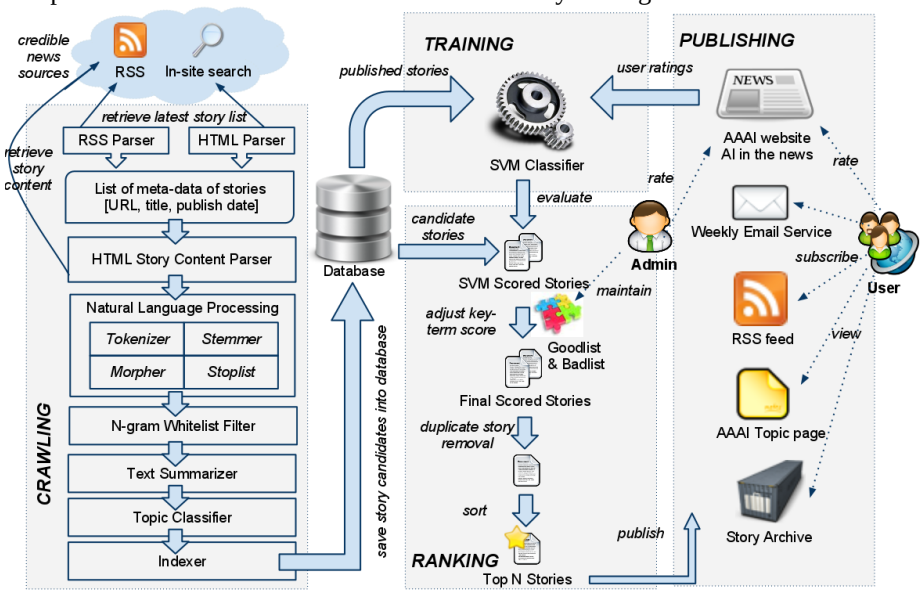

Fig. 1. NewsFinder Procedure Diagram

1.1 Crawling

In the crawling phrase, the program collects a large number of recent news stories about AI. Since crawling the entire web for stories mentioning a specific term like 'artificial intelligence' brings in far too many stories, we restrict the crawling to about two dozen major news publications. This makes a story more credible and more likely to interest an international audience. The system administrators (AI subject matter experts) maintain a list of news sources, chosen for their international scope, credibility, and stability. These include The BBC, The New York Times, Forbes, The Wall Street Journal, MIT Technology Review, CNET, Discovery, Popular Science, Wired, The Washington Post, and The Guardian. Others can be added to the list.

NewsFinder collects the latest news stories via either in-site search or RSS feeds from those sources and filters out blogs, press releases, and advertisements. If a source has a search function, then the program uses it to find stories that contain

‘artificial intelligence’ or ‘robots’ or ‘machine learning’. If a source has RSS feeds, then NewsFinder selects those feed labeled as ‘technology’ or ‘science’.

In order to parse the text to retrieve the content of candidate pages, we write a specific HTML parser for each news source to identify and extract the news content from its news web pages. The advantage of this method is precision in that it can accurately extract news text stories and eliminate advertisements, user comments, navigation bars, menus and irrelevant in-site hyperlinks. The disadvantage of writing separate parsers for each news source is somewhat offset by starting with a generic template. We have written a dozen specific source parsers as modifications of code inherited from a base parser. Each parser is specifically designed for one news source web site since different sites use different HTML/CSS tags. We are also investigating an alternative method [5, 10] a classification method is used to train parsers to recognize news content either by counting hyperlinked words or by visual layout.

For a typical news source the parser will extract three items from the metadata associated with each news item: URL, title, and publication date. If the publication date is outside the crawling period (currently seven days), the news story is skipped.[4] For the remaining stories, the parser extracts the text from of each story from its URL.

NewsFinder then processes the natural language text, using the Natural Language Toolkit (NLTK) [7] to perform word counting, morphing, stemming, and removal of the most common words from a stoplist.[5]

A text summarization algorithm extracts 4-5 sentences from the story to build a short description — the highlights that make the story interesting — since an arbitrary paragraph, like the first or last, is often not informative. The main idea of the algorithm is to first measure the term frequency over the entire story, and then select the 4-5 sentences containing the most frequent terms. In the end, it re-assembles the selected top 4-5 sentences in their original order for readability.

NewsFinder references a Whitelist of terms whose inclusion in a story is required for further consideration. If the extracted text contains no Whitelist term, the story is skipped. In addition to the term ‘artificial intelligence’, Whitelist includes several dozen other words, bigrams and trigrams that indicate a story has additional relevance and interest beyond the search term used to find it in the first place. For example, stories are retrieved from RSS feeds for the topic ‘robots’ but an additional mention of ‘autonomous robots,' or ‘unmanned vehicles’ suggests that AI is discussed in sufficient detail to interest AAAI readers.

The program then determines the main topic of each story. It uses the traditional Salton tf-idf cosine vector algorithm [8, 11] to measure the similarity of a story to the

---

[4] When using Google News, we also skip stories originating from a URL on our list of inappropriate domains. We set up the list initially to block formerly legitimate domains that have been purchased by inappropriate providers, but it can be used to block any that are known to be unreliable or offensive.

[5] The program also includes a Name Entity recognition algorithm, but it is not used routinely because it runs slowly. Instead, we check for names of particular interest, like “Turing”, by adding them to the Goodlist described in the Ranking section.

introductory pages of each of the major topics on the AITopics web site.[6,7] Each document is treated as a vector with one component corresponding to each term and its tf-idf weight. Thus, we can measure the similarity of two documents by measuring the dot product of their normalized vectors, which produces the cosine of the vectors' angle in a multi-dimensional space [8].

The story is then linked to the AITopics page with the highest similarity so that readers wanting to follow up on a story with background information on that topic. The story is also added to a list for the RSS feed for the selected topic. At publication time the topic is shown with the story and the RSS feed that contains it.

Finally, NewsFinder saves the candidate news stories and their metadata into a database for subsequent processing.

1.2 Training

In order to train NewsFinder's classifier to recognize stories that the readers of AITopics want to see, we collect ratings from them. The classifier is retrained periodically (currently every week), when an old set of stories is archived and a new set is about to be collected.

Readers are asked to rate the relevance and interest of a story for the AITopics readership as "not relevant to AI" (0), or 1-5 for a degree of relevance and interest.

The rating system is modeled after the five-star rating system used by Netflix [6], although our purpose is to classify unseen items with respect to their likely interest to other readers, and not just their interest to the specific individual doing the rating. While individualized suggestions may be added in the future, for now we assume that the aggregate of many ratings reflects the opinion of the community at large.

After each story we show the rating as in Fig. 2, including the average rating of other readers during the week, both as a number and as a row of stars, for readers who may wish to focus first on stories that others have rated highly.

Fig. 2. Rating Interface

The PmWiki Cookbook StarRater [1] is used to collect Users' ratings. We record each user's rating for every news story together with IP address and username. The IP

[6] The current major topics are: AI Overview, Agents, Applications / Expert Systems, Cognitive Science, Education, Ethical & Social, Games & Puzzles, History, Interfaces, Machine Learning, Natural Language, Philosophy, Reasoning, Representation, Robots, Science Fiction, Speech, Systems & Languages, Vision.

[7] The topic assignment algorithm was originally written in PHP by Tom Charytoniuk and rewritten in Python by Liang Dong.

address is a proxy for a user ID and allows us to record just one vote per news item per IP address.[8]

During training, all the readers' ratings are collected and averaged. If a news story has fewer ratings than a specified number, the average rating is ignored (unless it is from one of the administrators). If the standard deviation of a news story's ratings is greater than a cutoff (default 2.0), the ratings are discarded as well. This way, a news story is only added to the training set if there is general consensus among several raters about it (or if one of the administrators ranks it).

The Support Vector Machine (SVM) [3] is a widely used supervised learning method which can be used for classification, regression or other tasks by constructing a hyperplane or set of hyperplanes in a high dimensional space. An SVM from a python library LibSVM [4] has been trained with manually scored stories from the web to classify the goodness of each story into one of three categories: (a) high – interesting enough to *AI in the News* readers to publish, (b) medium – relevant but not as interesting to readers as the first group, and (c) low – not likely to interest readers. Currently, we build three 'one against the rest' classifiers to identify these three sets.

1.3 Ranking

After crawling and training, the next step is ranking the candidate stories during the current news period by computing and comparing the scores of all news stories crawled during the period. The score for each news story is computed in two steps: (i) assign an SVM score and (ii) adjust it using a key term score.

The SVM score is assigned based on which of the three SVM categories has the highest probability: high interest = 5, medium = 3, low or no interest = 0. If none of the classifiers assigns a 50% or greater probability of the story being in its category, the default score for the story is 1. The probability is based on the tf-idf measure of interest of all non-stop words in the document, typically about 200 words.

NewsFinder performs an adjustment to the SVM score by first retrieving every recent news story containing a term from a list called Goodlist. Terms on Goodlist are those whose inclusion in a story signals higher interest, as determined by subject-matter experts.

NewsFinder then measures the tf-idf score for each Goodlist term. All the term scores are accumulated and normalized across the recent stories.

When a new topic of interest first appears in AI, as "semantic web" did several years ago, the SVM can automatically recognize its importance as readers give high ratings to stories on this topic. Normal practice is for authors of stories on a new topic to tie the topic to the existing literature. However, an administrator (who is a subject matter expert) may also add new terms to Goodlist to jump-start this practice. Although one can imagine many dozen key terms on Goodlist, the initial two tests reported here used only 12 terms.

---

[8] As with Netflix, if there are multiple ratings for the same story from the same reader (more precisely, from the same IP address), only the last vote is used.

The same process is executed for terms on a list called Badlist. Terms on Badlist are those whose inclusion in a story signals lower interest. Initial testing was done using 12 Badlist terms. Both Goodlist and Badlist are easily edited in the setup file.

The key term score from Goodlist lies in [0, +1], which boosts the final score. The key term score from Badlist, which reduces the final score, is unbounded. Unlike the terms on Whitelist, whose omission forces exclusion of a story from further consideration, the terms on Goodlist and Badlist merely add or subtract from the initial SVM score based on the number of terms appearing and their frequency of occurrence. Multi-word terms on Goodlist, such as 'unmanned vehicle', have been manually selected as signals of increased interest. Badlist terms such as 'ceo', 'actor', and 'movie' can reduce the score for including unrelated news such as gossip about actors who appeared in Spielberg's movie "Artificial Intelligence." The terms 'tele-operated' and 'manually operated' similarly reduce the score on many stories about robots that are less likely to involve AI.

The computation of the key term score is as follows: given a key term such as 'automated robot', the program first finds all the recent stories containing both 'automated' and 'robot'. Then it computes the tf-idf score for each term, and adds all the tf-idf scores for this story.

After NewsFinder obtains the trained SVM score and key term score, each news story's final score is a weighted sum of its SVM score and its key term score, where the weight of the weight term, w, was selected heuristically to be 3.0:

$$Score = SVMScore + w \cdot KeyTermScore$$

Currently, both Goodlist and Badlist are manually maintained by the webmaster, in order to control quality during startup. When the size of the training set reaches about 500 stories, we plan to remove both lists.

It is worth-noting that the length of the story doesn't affect the SVM score since each story's tf-idf is normalized before being classified. But it affects the key term scores since each term's tf-idf depends on the number of terms in the document. However, longer stories are *prima facie* more likely to include more key terms. In addition, when selecting from among similar stories, the program prefers longer ones.

After all the potential candidates have been scored, NewsFinder measures the text similarity to eliminate duplicate stories. The program clusters all the news candidates to identify news about the same event. These may be exact duplicates (e.g., the same story from one wire service used in different publications), or they may be two reports of the same event (e.g., separately written announcements of the winner of a competition). Again, NewsFinder measures the cosine of the angle between the two documents' tf-idf to determine their similarity in the vector space. If the computed similarity value is greater than a cutoff (0.33 by default), these stories are clustered together. If there is more than one story in a group, the story with the highest final score is kept for publishing.

The N-highest-scoring stories are selected for publishing each week. At the current time, these are the N (or fewer) "most interesting" stories with final scores above a threshold of 3.0; i.e., ranked "medium" to "very high." For the initial testing, N=20; in the last test and current version, N=12.

1.4 Publishing

The stories selected for publishing are those with the highest final scores from the ranking phase, but these still need to be formatted for publishing in different ways: (i) posting summarized news stories on the Latest *AI in the News* page of the AITopics web site, (ii) sending periodic email messages to subscribers through the "AI Alerts" service, (iii) posting RSS feeds for stories associated with major AITopics, (iv) archiving each month's collection of stories for later reference, and (v) posting each news story into a separate page on the AITopics web site.

2 Validation

2.1 SVM Alone

After training on the first 100 cases scored manually, we determined the extent to which the selections of the SVM part of NewsFinder matched our own. For a new set of 49 stories retrieved from Google News by searching for 'artificial intelligence', we marked each story as "include" or "exclude" from the stories we would want published, and we matched these against the list of stories NewsFinder would publish, without use of the additional knowledge of terms on Goodlist and Badlist. On the unseen new set of 49 recent stories crawled from Google News, the SVM put 46 of 49 stories (94%) into the same two categories – include as "relevant and interesting" or exclude – as we did. Five stories would have been included for the 10-day period, which we take to be about right (but on the low side) for weekly email alerts.

This was not a formal study with careful controls since the person rating the stories could see the program's ratings, and the SVM was retrained using some of the same stories it then scored again. Nevertheless, it did suggest that the SVM was worth keeping. It also suggested that merely using an RSS feed or broad web search with a term like 'artificial intelligence' would return many more irrelevant and low-interest stories than we wanted. In a one week period Google News returned 400 candidate stories mentioning the term 'artificial intelligence', 88 mentioning 'machine learning', 8,195 mentioning 'robot', and 2,264 mentioning 'robotics.' We concluded that not all would be good to publish in *AI in the News*, nor would readers want this many.

2.2 Adjusted Scores

In a subsequent test, we used a specified set of credible news sources, a training set of 265 stories (including the 149 from the initial test), and a test set of 69 new stories. The full NewsFinder program was used, with scores from the SVM adjusted by additional knowledge of good and bad terms to look for. We compared the program's decision to include or exclude from the published set against the judgment of one administrator (BGB) that was made before looking at the program's score.

We accumulated scores and ratings by the administrator for 3-4 stories per day that were not in each previous night's training set, a total of 69 stories in the first three weeks of September, 2010. Although the SVM is improving (or at least changing)

each night, these stories are truly “unseen" in the sense that they did not yet appear in the training set used to train the classifier that scored them. Among 42 stories that the program scored above the publication threshold (≥ 3.0), the administrator rated 33 (78.6%) above threshold.

Out of 27 candidate stories that the program rated below the publication threshold (< 3.0), the administrator rated 11 (40.1%) below threshold. Thus the program is publishing mostly stories that the administrator agrees should be published but is omitting about half the likely candidates that the administrator rates above threshold. The 27 candidates in this study that were not published were mostly “near misses.” Many were rated 3 by the administrator, indicating that they were OK, but not great. Also, a few of the stories the administrator would have published may be selected on a later day, after retraining or when their normalized scores rise above threshold because the best story in the new set is not as good as in the previous set. Given a limit of twelve stories, the tradeoff between false positives and false negatives weighs in favor of omitting some good stories over including uninteresting or marginal ones.

We conducted a 5-fold cross validation for 218 stories with administrator ratings to validate the performance of the SVM classifier (before adjustment). As above, each of the tests was on “unseen” stories. For these 218 valid ratings, we counted the times that the administrator and the SVM classified a story in the same way. The accuracy of the “high” predictions was 66.5%, of the “medium” ratings 72.9%, and of the “low” ratings 74.3%.

2.3 Final Test

After the completion of these tests, some adjustments were made to correct occasional problems noted during testing.

- A story categorized as low or no interest by the SVM (category 0) is not published, regardless of its adjusted score.
- The threshold for similarity of two news stories was lowered from 0.4 to 0.33 to reduce the number of duplicates.
- Whitelist and Goodlist were made to contain the same terms, though their uses remain different. Thus a story must contain at least one of several dozen terms to be considered at all (Whitelist), and the more occurrences of these terms that are found in a story, the more its score will be boosted (Goodlist). Three new terms were added to Whitelist and Goodlist.
- Upward adjustments to the score from the key terms on Goodlist are now normalized to the highest adjustment in any period because adding a larger number of Goodlist terms created uncontrollably large adjustments. (Unbounded downward adjustments do not concern us because stories containing Badlist terms are unwanted anyway.)
- Terms having to do with tele-operated robots and Hollywood movies were added to the Badlist, thus downgrading stories that are about manually controlled robots or movie personalities.
- The frequency with which the program searches for stories and publishes a new *AI in the News* page was changed from daily to weekly.

- The number of stories published in any period has been changed from 20 to 12, since that will reduce the false positives and also reduce the size of weekly email messages to busy people.
- Stories can be added manually to be included in the current set of stories to be ranked. Thus when an interesting story is published in a source other than the ones we crawl automatically, it can be considered for publication. It will also be included in subsequent training, which may help offset the inertia of training over the accumulation of all past stories and the lag time in recognizing new topics.

A follow-up test was performed on 118 unseen stories to confirm that the changes we had made were not detrimental to performance. We also gathered additional statistics to help us understand the program's behavior better. Two-thirds of the stories were at or above the program's publication threshold of 3.0 (80/118), based on their initial SVM and adjustment scores).

Among 118 stories that passed the relevance screening and duplicate elimination, and thus were scored with respect to interest, the overall rate of agreement between the program and an administrator is 74.6% on decisions to publish or not (threshold ≥ 3.0), with Precision = 0.813, Recall = 0.813, and F1 = 4.92. Both the program and the administrator recommend publishing about two-thirds of the stories passing the relevance filters, just not the same two-thirds.

Table 1. Decisions on 118 Stories Rated by Both Admin and NewsFinder

	Admin:	Admin:
NewsFinder: Publish	65 (55%)	15 (13%)
NewsFinder: Don't Publish	15 (13%)	23 (20%)

3 Conclusions

Replacing a time-consuming manual operation with an AI program is an obvious thing for the AAAI to do. Although intelligent selection of news stories from the web is not as simple to implement as it is to imagine, we have shown it is possible to integrate many existing techniques into one system for this task, at low cost. There are many different operations, each requiring several parameters to implement the heuristics of deciding which stories are good enough to present to readers. The two-step scoring system appears to be a conceptually simple way of combining a trainable SVM classifier based on term frequencies with prior knowledge of semantically significant term relationships.

NewsFinder has not been in operation for long, but it appears to be capable of providing a valuable service. We speculate that it could be generalized to alert other groups of people to news stories that are relevant to the focus of the group and highly interesting to many or most of the group. The program itself is not specific to AI, but the terms on Goodlist and Badlist, the terms used for searching news sites and RSS feeds, and to some extent the list of sources to be scanned, are specific to AI.

Learning how to select stories that the group rates highly adds generality as well as flexibility to change its criteria as the interests of the group change over time.

Acknowledgments. We are grateful to Jim Bennett for many useful comments especially on the Netflix rating system after which the NewsFinder rating is modeled, and to Tom Charytoniuk for implementing the initial prototype of this system.

4 References

1. 5 star rating system for Pmwiki, http://www.pmwiki.org/wiki/Cookbook/StarRater
2. Buchanan, B. G., Glick, J. and Smith, R. G. 2008. "The AAAI Video Archive." *AI Magazine*, **29**(1): 91-94.
3. Burges, J.C. Christopher. "A tutorial on support vector machines for pattern recognition". Data Mining and Knowledge Discovery 2(2):121-167.
4. Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, (2001)
5. Suhit Gupta, Gail Kaiser, David Neistadt, Peter Grimm. DOM-based Content Extraction of HTML Documents. Proceedings of the 12th International World Wide Web Conference(WWW2003). Budapest, Hungary (2003)
6. J. Herlocker, J. Konstan, L.Terveen, J.Riedl, Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems 22, ACM Press, (2004)
7. E. Loper, S. Bird, NLTK: The Natural Language Toolkit, In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics, http://www.nltk.org/ (2002)
8. C. Manning, et.al. Intro. to Information Retrieval, Cambridge University Press, (2008)
9. P. Melville and V. Sindhwani. Recommender System. In Encyclopedia of Machine Learning, Claude Sammut and Geoffrey Webb (Eds), Springer, 2010
10. Ruihua Song, Haifeng Liu, Ji-Rong Wen, Wei-Ying Ma.. Learning Block Importance Models for Web Pages. Proceedings of the 13th International World Wide Web Conference (WWW 2004). New York. (2004)
11. G. Salton, Gerard and Buckley, C. Term-weighting approaches in automatic text retrieval. Information Processing & Management 24 (5): 513–523. doi:10.1016/0306-4573(88)90021-0, (1988)
12. Sheng Zhang, Weihong Wang, James Ford, Fillia Makedon, Learning from incomplete ratings using non-negative matrix factorization. In proc. Of the 6th SIAM conference on data mining, 2006.