Images in News

Jagan Sankaranarayanan Hanan Samet
University of Maryland
{jagan,hjs}@cs.umd.edu

Abstract

A system, called NewsStand, is introduced that automatically extracts images from news articles. The system takes RSS feeds of news article and applies an online clustering algorithm so that articles belonging to the same news topic can be associated with the same cluster. Using the feature vector associated with the cluster, the images from news articles that form the cluster are extracted. First, the caption text associated with each of the images embedded in the news article is determined. This is done by analyzing the structure of the news article's HTML page. If the caption and feature vector of the cluster are found to contain keywords in common, then the image is added to an image repository. Additional meta-information are now associated with each image such as caption, cluster features, names of people in the news article, etc. A very large repository containing more than 983k images from 12 million news articles was built using this approach. This repository also contained more than 86.8 million keywords associated with the images. The key contribution of this work is that it combines clustering and natural language processing tasks to automatically create a large corpus of news images with good quality tags or meta-information so that interesting vision tasks can be performed on it.

***Keywords*-News images; online clustering; image tags, news image corpus**

I. Introduction

We describe the extension of NewsStand [1] (denoting "Spatio-textual Aggregation of News and Display"), a system for the aggregation of news articles from RSS feeds into clusters (see also TwitterStand [2] which works with news tweets) to automatically extract images from news and enable the retrieval of similar images by use of natural language specification techniques. The result is a search engine that can retrieve similar images from a repository of news articles. In particular, when fully implemented, the search engine can be used to answer queries such as — "Find an image with John Wayne riding a horse backwards", or "Find an image with Alexander Solzhenitsyn playing tennis". Ideally, we would like to use computer vision algorithms to achieve such image retrieval tasks. Unfortunately, such a technology is still at least a decade or more away. However, we can circumvent some of the hard computer vision issues associated with such image retrieval tasks by examining the captions associated with the images. In other words, if we know that the image is of "John Wayne" and a "horse", and if the caption contain the keywords "riding backwards", then we can retrieve such an image. So, relevant and appropriate keywords associated with an image mean that their knowledge can be used to help us perform interesting image retrieval tasks. In this paper, we concentrate our effort on the news domain due to its wide availability and its richness in terms of images. The challenge becomes one of how to automatically collect a large repository of news images and associate them with appropriate *keywords*. In particular, we want to associate images with a caption which is a succinct description of what is shown in the image, what action is being performed (e.g., Solzhenitsyn playing tennis), keywords, a small pool of possible people who may be present in the image, and locations or organizations related to it. We argue that if one were to build such a large image repository, then interesting computer vision problems could now be addressed. For example, if a query looks for "Solzhenitsyn" in an image, we can restrict all those images where "Solzhenitsyn" is present in the list of people associated with the image, making the problem a little easier. We can then automatically learn *appearance* model for Solzhenitsyn by clustering all images based on the features drawn from any *face* in the images. The cluster centroid forms the appearance model for Solzhenitsyn, which means that we can now use it to pick out other pictures containing Solzhenitsyn. As one can see, our idea of automatically generating good quality keywords for a large image corpus can be used in building interesting computer vision applications.

In this paper, we describe a method to automatically create a large image corpus from news articles. The

This work was supported in part by the National Science Foundation under Grants IIS-09-48548, IIS-08-12377, CCF-08-30618, and IIS-07-13501, as well as NVIDIA Corporation, Microsoft Research, and Google.

method we propose takes several news feeds as inputs and then automatically extracts the images contained within them. Furthermore, the algorithm can associate a set of keywords, people's name, locations, and organizations that are relevant to what has been captured by the image. The first step in this process is to cluster documents so that we can associate the document with a set of keywords (called feature vector). Next, using Natural Language Processing (NLP) analysis, we determine a set of people, locations, and organizations that are present in the news article using techniques that we first developed for the STEWARD system [3], [4]. Finally, we use the feature vector to determine which of the images in a news article is relevant to the content of the news story. Note that a news article may contain many images unrelated to the news story, such as icons, placeholders, advertisements, etc. Using our approach allows us to quickly create large image repositories with extremely good quality tags or keywords associated with each image, which is important to several emerging applications in computer vision.

II. Preliminaries

Clustering algorithms have been the subject of intense study [5] for quite some time. One common document clustering strategy is to first convert the documents to a *feature vector* [6] representation using the *TF-IDF* [7] measure. These feature vectors, which are points in a very high-dimensional space, are then clustered using a simple distance function such as the *cosine* similarity measure [5]. If two such feature vectors are within distance of ϵ of each other, then they are said to be sufficiently similar to likely refer to the same news story. The similarity search can be done with indexed [8] or vector space embedding [9] methods.

A clustering algorithm for the news domain should group together all news articles that describe the same *news event* into groups of articles termed *story clusters*. Broadly, a news event is defined in terms of both *story content* and *story lifetime* — articles in the same cluster should share much of the same important keywords, and should have temporally proximate dates of publication. Time is an essential part of grouping news articles, since two articles may contain similar keywords but describe vastly different news events. For example, two stories about separate earthquakes may share many keywords, but should be placed in separate clusters if one story was breaking news and the other was several days old. Additionally, we want new or breaking articles to be clustered quickly, so that breaking stories can be presented immediately to users.

This speed requirement precludes the use of traditional one-shot approaches to clustering. For every new article downloaded, the entire news collection would have to be clustered again, incurring unacceptable performance penalties for voluminous news days. Instead, we take an *incremental* or *online* approach to clustering that reuses existing clusters, and requires significantly less computation time. Furthermore, we use the above temporal constraint and several optimizations to effect real-time processing of thousands of articles per day.

We use the *vector space model* [6] of documents, often used in text mining and information retrieval. This model represents a text document as a *term feature vector* in a d-dimensional space, where d is the number of distinct terms in every document in a corpus. Note that the term feature vector is distinct from the entity feature vector. Each element of the term feature vector represents the frequency of its corresponding term in the document, as computed by a term weight formula. d will evolve as articles are added and removed from the space, which must be accounted for in the online clustering. Furthermore, the vector space is usually high-dimensional, with typical d values of 100,000 or more, so ordinary $O(d)$ distance computations can be prohibitively expensive. However, we take advantage of the *sparseness* of these term feature vectors to expedite distance computations and achieve good performance. Our methods for computing term feature vectors and clustering are described in further detail below.

III. Feature Extraction

Upon receiving a new article to be clustered, we first normalize the article's content by *stemming* [10] input terms and removing punctuation and other extraneous characters. Next, we extract the article's term feature vector by computing the well-known *Term Frequency-Inverse Document Frequency* (TF-IDF) [7] score for each term in the article. This score emphasizes those terms that are frequent in a particular document and infrequent in a large corpus D of documents. The TF-IDF score for a term t_i in article d_j is

$$\text{TF-IDF}_{i,j} = \frac{n_{i,j}}{n_j} \cdot \log \frac{|D|}{O_i}$$

where $n_{i,j}$ is the number of occurrences of t_i in d_j, n_j is the number of terms in d_j, and O_i is the number of articles in D that contain t_i. For our corpus, we simply use the collection of news articles present in our clustering. Note that even though our corpus constantly evolves with each new article processed, for performance reasons, we compute the term feature vector for a particular article only once, upon its addition to the system. In practice, this optimization does not affect clustering noticeably.

IV. Online Clustering

Our clustering algorithm is a variant of leader-follower clustering [11] that permits online clustering in both the term vector space and the temporal dimension. For each cluster, we maintain a *term centroid*

and *time centroid*, corresponding to the means of all term feature vectors and publication times of articles in the cluster, respectively. To cluster a new article a, we check whether there exists a cluster where the distance from its term and time centroids to a is less than a fixed cutoff distance ϵ. If one or more candidate clusters exist, a is added to the closest such cluster, and the cluster's centroids are updated. Otherwise, a new cluster containing only a is created.

We use a variant of the *cosine similarity measure* [5] for computing term distances between the new article and candidate clusters. The term cosine similarity measure for a article a and cluster c is defined as

$$\delta(a,c) = \frac{\overrightarrow{TFV_a} \bullet \overrightarrow{TFV_c}}{||\overrightarrow{TFV_a}||\,||\overrightarrow{TFV_c}||}$$

where $\overrightarrow{TFV_k}$ is the term feature vector of k.

In order to account for the temporal dimension in clustering, we apply a Gaussian attenuator on the cosine distance that favors those clusters whose time centroids are close to the article's publication time. In particular, the Gaussian parameter takes into account the difference in days between the cluster's time centroid and the new article's publication time. Our modified distance formula is

$$\dot{\delta}(a,c) = \delta(a,c) \cdot e^{\frac{-(T_a - T_c)^2}{2(2.2)^2}}$$

where T_a is a's publication time and T_c is c's time centroid.

To improve performance, we store cluster centroids in an inverted index that contains, for every term t, pointers to all clusters that have non-zero values for t. We use this index to reduce the number of distance computations required for clustering. When a new article a is clustered, we compute the distances only to those clusters that have non-zero values in the non-zero terms of a. As a further optimization, we maintain a list of *active* clusters whose centroids are less than a few days old. Only those clusters in the active list are considered as candidates to which a new article may be added. We remove clusters from the active list after several days, since the values from our distance function will be negligible. Together, these optimizations allow our algorithm to minimize the number of distance computations necessary for clustering articles.

V. Cluster Feature

The identification of entities (i.e., people, location, and organization) in news articles is facilitated by the use of a Natural Language Processing (NLP) method known as *Named-Entity Recognition* (NER) [12]. Existing NER taggers use a variety of techniques from statistical learning [13], [14], [15], [16], [12] and natural language processing [17], [18], [19], as well as hybrid approaches [20]. Our NER tagger is based primarily on LingPipe [21] with customization for the news domain. Our NER tagger takes a news article as input and annotates words and phrases in the news articles of location, people and organization. Now, we can aggregate on all the articles belonging to a cluster to obtain the most common location, people, and organization names mentioned in all the articles belonging to the cluster. In other words, if an entity appears in a majority of the articles belonging to a cluster, then we can assume that it is important to the news topic. By requiring that an entity appears in most of the articles associated with a cluster, we can average out most of the entities that are noise resulting in good quality output. Note here that the feature vector corresponding to the cluster centroid, caption (described below) and the entities are all associated with each image that is extracted from an article.

VI. Image Extraction

Now that an article has been clustered, we can extract all the relevant images from it using the feature vector of the cluster. Image extraction from a web page requires processing of the HTML tags so that a caption can be associated with each of the images in the news article. The caption is a textual description of the image, which usually describes the scene captured by it. In most cases, it may also include the people present in the image. Observe that we may not be able to identify a caption for each of the images in a news article, but from our experience, we are able to do so for a large percentage of them. Note also that the caption of an image is usually not very descriptive. However, it does succinctly capture both the content of the news article as well as the scene captured by the image. So, the caption of the image is *similar* to the content of the article, which forms the central idea of this paper.

We examine every image in the HTML page. If we can visualize the HTML as a tree structure, and the image as a node in the tree, then the idea is to look at the children nodes and a few ancestor nodes to try to collect enough text which would serve as the caption of the image. Sometimes the image may have a TITLE field associated with it, in which case it forms the caption of the image. In some cases there may be an ALT field, which can also serve as a caption. We also look for configurations where the image is embedded in a DIV element, in which case we use any text found within the DIV element. Our algorithms use several configuration such as nested DIV and TABLE structures, or combinations of them. Note that the caption is usually not very long in length, which means that we can simply discard any text if it is too long. In addition, we note that we will also require that the image have a minimum size and that it has an aspect ratio greater than 1.5 as is typical with images accompanying news articles.

Now, that we have a caption for the image, we try to match the cluster feature to see how many keywords from the cluster feature are found in the description. For example, if the feature vector of the document contains "Hurricane", "Katrina", "Bush", "Brown", and "FEMA", then we would expect one or more of these features to be present in the caption text. If not, then we simply discard the image. Once the image has been extracted, we also record the caption text, the cluster feature, the names of the people and organization with the image.

VII. Experimental Results

We applied our method to a data-set containing 12 million news articles. Our algorithm extracted more than 983k images from this data-set. Our tagging and clustering methods resulted in associating these images with 86.8 million keyword terms, 12.3 million names of people, and 6.4 million geographical locations. Figure 1 shows a few sample images from the news story (i.e., cluster of relevant articles) on the January 2010 earthquake in Haiti. We examined 1261 images from this story of which all were found to be relevant to the earthquake. The feature vector for the cluster contained the following keywords "Earthquake", "Haiti", "Port", "UN", "Food", "Aid", "Survivor", "Rescue" "Tent", "Telethon" and "Relief".

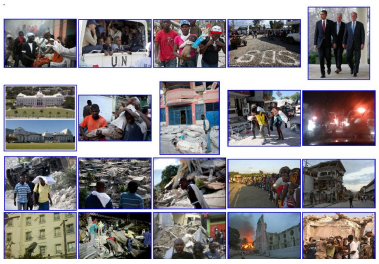

Fig. 1. A few sample pictures from the news story on the earthquake in Haiti.

VIII. Conclusion

We presented a method for automatically extracting images from news articles using an online clustering algorithm and natural language-based techniques. Using this approach, we created a corpus containing 983k images with more than 86 million keywords tags for it. This work opens up the possibility of applying computer vision algorithms to large image repositories. The NewsStand system is available http://newsstand.umiacs.umd.edu/news as is the accompanying TwitterStand system.

References

[1] B. Teitler, M. D. Lieberman, D. Panozzo, J. Sankaranarayanan, H. Samet, and J. Sperling, "NewsStand: A new view on news," in *Proceedings of ACM SIGSPATIAL GIS*, Irvine, CA, Nov. 2008, pp. 144–153.

[2] J. Sankaranarayanan, H. Samet, B. Teitler, M.D. Lieberman, and J. Sperling, "TwitterStand: News in tweets", in *Proceedings of ACM SIGSPATIAL GIS*, pp. 42–51, Seattle, WA, Nov. 2009.

[3] M. D. Lieberman, H. Samet, J. Sankaranarayanan, and J. Sperling, "STEWARD: architecture of a spatio-textual search engine," in *Proceedings of ACM GIS*, pp. 186–193, Seattle, WA, Nov. 2007.

[4] H. Samet, M. D. Lieberman, J. Sankaranarayanan, and J. Sperling, "STEWARD: Demo of spatio-textual extraction on the web aiding the retrieval of documents," in *Proceedings of the 7th National Conference on Digital Government Research*, pp. 300–301, Philadelphia, PA, May 2007.

[5] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *KDD Workshop on Text Mining*, Boston, MA, Aug. 2000, pp. 1–20.

[6] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," in *CACM*, vol. 18, no. 11, pp. 613–620, 1975.

[7] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[8] R. J. Bayardo, Y. Ma, and R. Srikant, "Scaling up all pairs similarity search," in *Proceedings of World Wide Web*, May 2007, pp. 131–140.

[9] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, Edinburgh, Scotland, Sep. 1999, pp. 518–529.

[10] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[11] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley Interscience, 2000.

[12] G. Zhou and J. Su, "Named entity recognition using an HMM-based chunk tagger," in *Proceedings of the Annual Meeting on Association for Computational Linguistics*, Philadelphia, PA, 2001, pp. 209–219.

[13] J. D. Burger, J. C. Henderson, and W. T. Morgan, "Statistical named entity recognizer adaptation," in *Proceedings of the Conference on Natural Language Learning*, Taipei, Taiwan, Aug. 2002, pp. 163–166.

[14] R. Malouf, "Markov models for language-independent named entity recognition," in *Proceedings of the Conference on Natural Language Learning*, Taipei, Taiwan, Aug. 2002, pp. 187–190.

[15] P. McNamee and J. Mayfield, "Entity extraction without language-specific resources," in *Proceedings of the Conference on Natural Language Learning*, Taipei, Taiwan, Aug. 2002, pp. 183–186.

[16] D. Wu, G. Ngai, M. Carpuat, J. Larsen, and Y. Yang, "Boosting for named entity recognition," in *Proceedings of the Conference on Natural Language Learning*, Taipei, Taiwan, Aug. 2002, pp. 195–198.

[17] S. Cucerzan and D. Yarowsky, "Language independent NER using a unified model of internal and contextual evidence," in *Proceedings of the Conference on Natural Language Learning*, Taipei, Taiwan, Aug. 2002, pp. 171–175.

[18] Y. Ravin and N. Wacholder, "Extracting names from natural-language text," IBM Research Report, Yorktown Heights, NY,, Tech. Rep. RC 2033, 1997.

[19] D. A. Smith and G. Crane, "Disambiguating geographic names in a historical digital library," in *Proceedings of the European Conference on Research and Advanced Technology for Digital Libraries*, Darmstadt, Germany, 2001, pp. 127–136.

[20] J. Patrick, C. Whitelaw, and R. Munro, "SLINERC: the Sydney language-independent named entity recogniser and classifier," in *Proceedings of the Conference on Natural Language Learning*, Taipei, Taiwan, Aug. 2002, pp. 199–202.

[21] B. Baldwin and B. Carpenter, "Lingpipe," http://alias-i.com/lingpipe/, retrieved Jan 25, 2010.