Large-Scale Multimedia Content Analysis Using Scientific Workflows

Ricky J. Sethi
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
rickys@sethi.org

Yolanda Gil
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
gil@isi.edu

Hyunjoon Jo
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
hyunjoon@usc.edu

Andrew Philpot
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
philpot@isi.edu

ABSTRACT

Analyzing web content, particularly multimedia content, for security applications is of great interest. However, it often requires deep expertise in data analytics that is not always accessible to non-experts. Our approach is to use scientific workflows that capture expert-level methods to examine web content. We use workflows to analyze the image and text components of multimedia web posts separately, as well as by a multimodal fusion of both image and text data. In particular, we re-purpose workflow fragments to do the multimedia analysis and create additional components for the fusion of the image and text modalities. In this paper, we present preliminary work which focuses on a Human Trafficking Detection task to help deter human trafficking of minors by thus fusing image and text content from the web. We also examine how workflow fragments save time and effort in multimedia content analysis while bringing together multiple areas of machine learning and computer vision. We further export these workflow fragments using linked data as web objects.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Social Media Analysis; Multimodal Information Fusion; Scientific Workflows; Human Trafficking Detection; Big Data

1. INTRODUCTION

Analyzing multimedia content on the web is of great interest but it often involves leveraging expertise in data analysis, as well as social media [1, 2]. This is especially apparent in detecting human trafficking on the web as trafficking of minors via the internet is a fast-growing security problem [3]. Traffickers of minors are increasingly using internet-based ads and social media to advertise their illicit wares. These ads consist of both text and image descriptions of people, some of whom are minors being trafficked.

In this paper, we present preliminary work which focuses on the task of Human Trafficking Detection (HTD). We present our initial work to help analyze posts on various sites on the internet, which add approximately 20GB of posts every day, in order to determine if the subjects of these posts are victims of human trafficking. The ultimate goal of the project is to create intelligence which may be used by law enforcement to detect and combat such trafficking of minors.

This complex task requires examination of both text and image information, as well as the fusion of these two distinct data domains. Our approach is based on scientific workflows, which capture end-to-end methods that combine fragments developed by different people with expertise in different aspects of the task. Workflows consolidate heterogeneous codebases and programs written in many different languages [4, 5, 6]. Such workflows, designed by domain experts in their own fields, may also be of great utility to scientists in other disciplines. The Wings workflow system, in particular, was developed to assist scientists in managing complex computations [7, 8] and it has been used in several large-scale distributed scientific applications. For this work, we re-purposed existing workflows [9, 10, 11] and developed new workflows for image processing and fusion of image and text analyses.

Contributions: We thus use semantic workflows which capture expert-grade text and image analytics. Using the Wings framework, we have started initial development of a prototype which examines both the text of the post, as well as the associated images, in order to make a determination of the age of the person represented in the ad. We are further extending this semantic workflows-based approach to sup-

MM'13, October 21–25, 2013, Barcelona, Spain.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2404-5/13/10 ...$15.00.
http://dx.doi.org/10.1145/2502081.2502082.

port re-usability of workflow fragments and reproducibility of scientific research.

The main contribution of this paper is the re-use and repurposing of workflow fragments to facilitate rapid development of applications and to bridge expertise across disciplines when analyzing multimedia web content. We introduce several workflow fragments for Text Analytics and Image Analysis, as well as for the fusion of these different data modalities. We also use linked data principles to export these workflow fragments as web objects for the research community. In particular, we show how a social media analysis task can be solved rapidly and extended to do a multimedia analysis, which involves both text and image analytics, by simply re-using the workflow fragments developed by us and by others.

The rest of this article is organized as follows: in Section 2, we provide some background on the initial development of the HTD project. Then, in Section 3, we discuss the Wings workflow system. In Section 4, we show several workflow fragments we created for Text and Image Analytics, as well as their fusion. Then, in Section 5, we show how the re-use of workflow fragments can extend this nascent social media analysis task. Next, in Section 6, we show the results of experiments and give details of our implementation. Finally, in Section 6.5, we highlight the time and work savings allowed by the re-use of workflows, followed by a discussion of future directions in Section 7.

2. HUMAN TRAFFICKING DETECTION

Human trafficking of minors for the purposes of illicit, commercial exploitation is a serious international crime. Increasingly, traffickers use Internet classified ads, bulletin boards, and social media to advertise the illicit services of the persons they offer [3]. A very common scenario is to have advertisements for trafficked persons, both adults and minors, injected into classified advertisement listings for licit and quasi-licit personal services. In order to attract clients online, such ads typically contain several of: phone numbers and/or other contact information; provider names (or pseudonyms); location(s) served; descriptions of provider race, physical characteristics, and purported age; suggested services and prices; and images. At the same time, providers try to avoid apprehension by law enforcement by omitting, falsifying, or obscuring information believed to be useful in apprehension, inclusion of deliberate features (or noise) believed to insulate against law enforcement action, and by changing details over time (e.g., multiple aliases) to defeat attempts to link separate ads together.

Accordingly, law enforcement efforts to combat human trafficking of minors seek to use the rich information source ads represent as a source of evidence. To do so effectively, one needs to:

1. Extract content from individual ads
2. Classify ad content as to whether the provider is trafficked into prostitution or engaged in some other activity and
3. Estimate provider true age (especially likelihood of underage participation). To assist in targeting apprehension, it is essential to
4. Determine the service modality the ad offers, whether "outcall" (provider travels to customer), "incall" (customer travels to provider), the setting, or agency referral, or other.
5. Records from individual ads so gathered must be linked together, a form of entity resolution [12], yielding a set of single-provider time profiles.

Information from third-party sites, such as missing child databases or recent nearby apprehension evidence, can be correlated with extracted information, when available, typically during this phase. These profiles are then cross-matched to suggest inter-provider relationships (e.g., co-advertised or shared phone) which, in turn, particularly when combined with provider mentions in certain discussion groups begins to populate a *social network*, whose members include providers, customers and procurers.

The TrafficBot project was initiated in late 2011 to address these information extraction needs for federal law enforcement. Initial efforts focused on identifying suitable classified ad listings, developing site-focused web crawlers, and developing site-specific extraction scripts. By mid-2012, the resulting archive, spanning 110 metropolitan markets (themselves composed of several hundred city-level crawl results), and had grown to nearly 500,000 harvested posts spanning an estimated 200,000 providers.

The current system with basic archive query capability supports some basic law enforcement goals. For example, users can filter ads from market M1 where reported age is less than 25 and where at least one reported phone number is tagged in market M2 (indicating likely provider movement).

Development of deeper data analytics, however, has lagged behind the collection results, especially since this task requires massive multimodal information fusion. For tasks such as age estimation and modality tagging, competing hypotheses include using supervised and unsupervised learning and clustering methods, across various proposed text, image, and network features. Moreover, various parameters and feature extraction schemes for each method need to be identified and tuned. The large search space, combined with an enormous and fast-growing data store, necessitates a systematic engineering approach.

The use of workflows offers notable advantages:

1. Support for natural modularization of the task space;
2. Ability to scale up via parallelization;
3. Access to a large and mature workflow fragment library to address many of the pending analytic needs;
4. Allowing incorporation of pre-existing data and existing code as fragments, as well as simple integration with an existing system; and
5. Ease of experimenting with workflows to set up various parameters and explore multiple scenarios quickly.

3. WINGS SEMANTIC WORKFLOWS

Our approach uses the Wings workflow system [13], which has three key features that make workflows accessible to users: a simple dataflow structure, an easy-to-use web interface, and an ability to export workflows and workflow fragments as web objects. This framework allows us to structure computer vision and machine learning tasks as computational workflow fragments described in high-level declarative notations and capable of processing large quantities of data that comes from multiple sources or files [6, 14]. Wings is open source, built upon open web standards from the World Wide Web Consortium (W3C), and is available at http://www.wings-workflows.org/.

A unique feature of Wings is that its workflow representations incorporate semantic constraints about datasets and workflow components. Wings reasons about dataset properties and component constraints to create and validate workflows and to generate metadata for new data products. Wings uses Pegasus as the execution engine for large-scale distributed workflow execution. Wings allows users to express high-level descriptions of their analysis goals, and assists them by automatically and systematically generating possible workflows that are consistent with that request. Users can be assisted in an interactive mode, where Wings generates suggestions and validates their inputs, or in an automatic mode, where Wings can elaborate their initial request and present the user with execution-ready workflows as options.

Using a semantic workflow system like Wings to assist with the design of such computational experiments allows for creating structured, systematic experiments that can be automated, thus allowing anyone to reproduce results, regardless of how parameters and thresholds are optimized for specific datasets. In addition, the Wings workflow system has an open modular design and can be easily integrated with other existing workflow systems and execution frameworks to extend them with semantic reasoning capabilities.

The Wings workflow framework thus has the ability to easily incorporate new algorithms and new data sources. In addition, the framework is self-adaptive in that it automatically selects methods and parameter values that are appropriate for the user's data and can analyze its current knowledge, detect missing information, attempt to induce it, and, when this is not possible, assess the value of seeking that missing information [15].

In fact, the Wings workflow system is pre-equipped with several expert-quality workflows that represent a powerful set of analytic methods [16]. It includes workflow fragments for general machine learning packages like Weka [17], document clustering packages like CLUTO, etc. We extend these repositories by creating workflow fragments based on popular computer vision [18, 19, 20, 21, 22] and machine learning packages like OpenCV [23], a standard computer vision library, and MALLET [24], a standard package for statistical processing and information extraction, as well as adding custom implementations of some state-of-the-art computer vision/machine learning models.

These packages have vastly heterogeneous implementations but the workflow fragments encapsulate the software with interfaces described by data types in the workflow system to make them reusable in different workflows. Wings ensures that only the right components are used in workflows by checking the semantic constraints of the input and output types for every component. The system ensures that only workflows with valid combinations of components are executed. The framework also includes several widely used datasets used for comparison purposes in the text analytics and computer vision community.

In addition, these workflow fragments can be exported in Wings by publishing them as web objects using Linked Data principles [25] and can be made available as part of a workflow library. These web objects, represented in RDF, allow direct access via unique URIs to workflow fragments or workflows, their components, and their associated datasets. Such web objects can then be imported into any workflow system that is compatible with the standard Open Provenance Model for workflow publication [25] so that other researchers can directly re-use or re-purpose any single workflow fragment or entire workflows.

4. WORKFLOW FRAGMENTS FOR MULTIMEDIA ANALYSIS

Workflows are usually composed of workflow fragments that are reused across workflows. Such predefined workflow fragments make complex analytics expertise readily available to new users. The components that make up workflow fragments can be written in heterogeneous languages: e.g., some components are in Java, others in matlab, and still others in C++ but the language of choice is irrelevant as the components are integrated into the workflows without reliance upon their individual implementation idiosyncrasies. This is possible because each individual program is converted into a workflow component via a short wrapper shell script (usually 3-5 lines of code) thus allowing any pre-existing program to be incorporated as a new component in a workflow or workflow fragment.

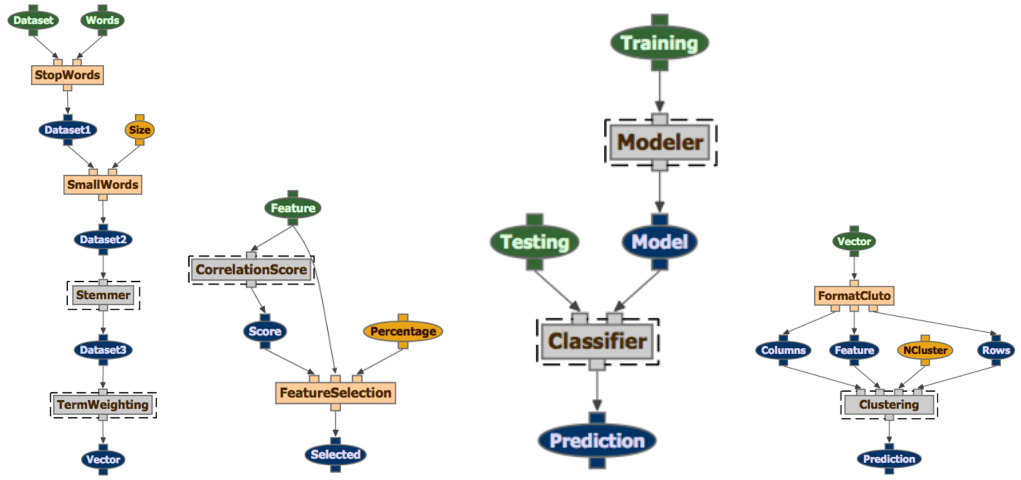
These previously defined workflow fragments can be executed independently from each other. This is helpful as some researchers might choose to focus on particular parts in order to optimize or improve their understanding of the behaviour in the individual steps. A good starting point for researchers in other disciplines, however, is to create end-to-end workflows that are formed by re-using and re-purposing workflow fragments. These end-to-end workflows would then incorporate and represent advanced expertise in that they would capture complex combinations of components that are known to work well in practice. Such re-usable workflow fragments are pre-defined by domain experts and available as part of workflow libraries. They can be executed with available datasets or adapted by adding or changing components.

Here, we detail some of the workflow fragments previously developed [9] for Text Analytics as seen in Figure 1 as well as worfklow fragments we have developed for Image Analytics in Figure 2.

4.1 Text Pre-Processing and Feature Generation

Analytic tasks usually begin with some preprocessing steps to generate the features of a document. The workflow fragment for feature generation is shown in Figure 1(a). Morphological variations are removed from the dataset with a stemmer component. The Wings workflow system provides several choices, including a Porter Stemmer and a Lovins Stemmer. It further provides term weighting components that is used to transform the dataset into the vector space model format. Among them are term frequency-inverse document frequency, corpus frequency or document frequency for instance. The generated outcome can now be used with different other workflows and is independent of a particular implementation at this stage in the workflows.

4.2 Feature Selection



(a) Generation (b) Selection (c) Classification (d) Clustering

Figure 1: Workflow Fragments previously developed [9] for Text Analytics. Here we see workflow fragments for a) Feature Generation; b) Feature Selection; c) Training and Classification; and d) Clustering.

A very common step for many data analytics tasks such as classification and clustering is feature selection, as shown in Figure 1(b), whose main purpose is to reduce the training set by only using the most valuable features. This will reduce the necessary time for training the model and can improve the results of the classifier in some cases. The goodness of a feature in the dataset is measured with correlation scores among the features. Typical implementations for this step are Chi Squared, Mutual Information or Information Gain that can be found in [26] and are all implemented in the framework. The resulting score is used in a feature selection step to retain the most valuable features in the training set. The percentage of selected features is typically changing for every dataset respectively classifier used in the computational experiment.

Another characteristic for this workflow fragment is that it uses heterogeneous implementations for the components. While the components for the computation of the correlation score take advantage of the capabilities of MATLAB to handle large matrices very elegant, the component for the feature selection uses an implementation written in Java.

4.3 Classification

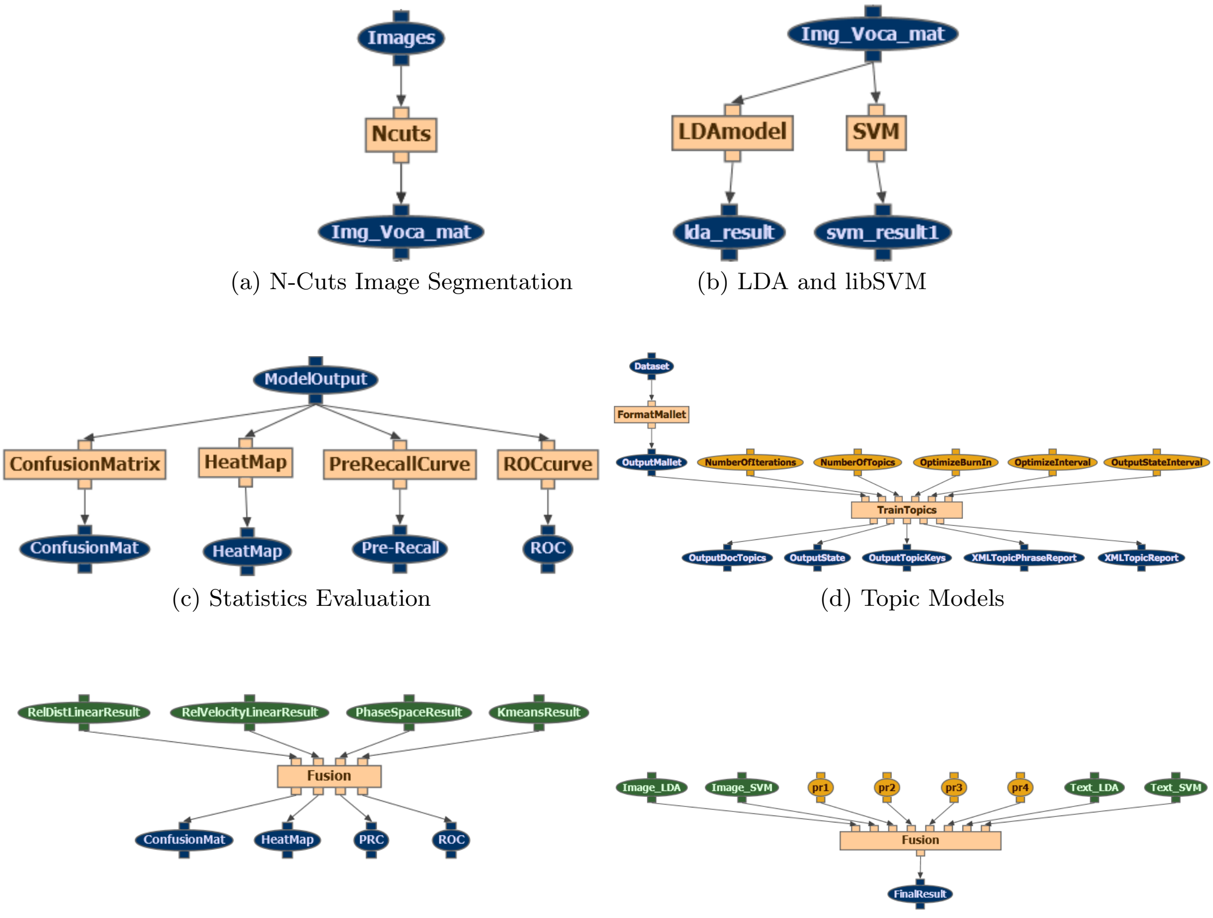
The resulting training set after the feature selection can be used for the training of a model with the workflow fragment shown in in Figure 1(c). Both components in the workflow use the Weka machine learning framework. Thus, many different machine learning algorithms can be used to perform experiments with the dataset. Among them are very popular algorithms from the text analytic community like Support Vector Machines, Naive Bayes or k-Nearest Neighbor. The computed model can be stored in the data catalog and reused for later classifications. Since the training is usually a very time demanding task in the workflows, it is very desirable to reuse previously created models. Existing models are also easier to compare against each other, because the metadata information of the model carries provenance information from the components used and their configuration during the workflow execution. In the second step a classifier uses the trained model with the testing set to compute the predictions.

4.4 Clustering

In Figure 1(d), we see the workflow fragment for clustering. The Vector that results from the Feature Generation workflow fragment in Figure 1(d) can be used as input for clustering. It needs to be formatted into the suitable format for the clustering software. The result of this step is the Feature output with the transformed Vector. Next to this output there are additional intermediate files called Rows and Columns that contain the label names that are used to annotate the final result with the right names for the features and labels. The parameter for this component is used to specify the number of clusters to be applied on the data set.

4.5 Image Analysis

Here, we detail some of the Workflow Fragments we have developed for Image Analysis as seen in Figure 2. In particular, we created workflow fragments for a) Normalized Cuts Image Segmentation [27] which views image segmentation as the optimal partitioning of a graph by minimizing the cut with a modified cost function; b) Latent Dirichlet Allocation (MALLET) [24] for visual-word clustering and Support Vector Machines (libSVM) for visual-word classification [28, 29]; c) Statistics Evaluation (Confusion Matrices, Heatmaps, Precision Recall Curves, and Receiver Operating Characteristic Curves) [18, 19, 30]; and d) Topic Modelling (MALLET) [24] for video-word clustering. Visual-words and video-words are the image and video equivalent of text words used in textual bag-of-words models; in computer vision, they are created by partitioning an image or video into interest point cuboids or segments and then computing some features (for which it is possible to calculate a distance metric) for each interest point cuboid. The centers of each of these clusters are the visual words (codewords) in the visual vocabulary (codebook). The statistics evaluation workflow fragment allows for easy visualization of diverse summary



(a) N-Cuts Image Segmentation

(b) LDA and libSVM

(c) Statistics Evaluation

(d) Topic Models

(e) FUSION: weighted average of all other methods

(f) Alt. FUSION module with user-controlled weights

Figure 2: Workflow Fragments for Image Analysis. Here we see workflow fragments for a) N-Cuts Image Segmentation; b) Clustering and classification through Latent Dirichlet Allocation (mallet) and Support Vector Machines (libSVM); c) Statistics Evaluation (Confusion Matrices, Heatmaps, Precision Recall Curves, and Receiver Operating Characteristic Curves); d) Topic Modelling (mallet); e) FUSION: a weighted average of four input methods with normalized output scores; and f) Alternate FUSION module for the HTD.

and graphical statistical measures which are the outputs of that component (i.e., summary measures like Equal Error Rate, Mean Average Precision, etc., as well as the graphical outputs of Confusion Matrices, Heatmaps, Precision Recall Curves, and Receiver Operating Characteristic Curves).

5. EXTENDING HUMAN TRAFFICKING DETECTION VIA WORKFLOWS

The initial development of the project had progressed to creating a crawler, which downloads posts from various posting sites, and an extractor, which extracts the text and images and stores them in a database. However, there had been no substantial analysis of the posts in this nascent project. We extended the project to examine both the text of the post (using the Text Analytics workflow fragments we had already developed), as well as the associated images (using the Image Analysis workflow fragments we developed); a final determination about trafficking of the subject of the post was made by fusing the results of the Text and Image analysis via the Fusion workflow fragment we developed. The goal of this project is to use both the text and image content of posts to make a stronger determination of whether or not the subject of the post was trafficked. Thus, the re-use and re-purposing of workflow fragments allowed a multimedia analysis spanning data domains of text and image analysis, including the fusion of their results in the final determination.

We extended the nascent HTD project by re-using and re-purposing Image Analysis and Text Analytics workflow fragments. In particular, we componentized, re-used, and re-purposed the workflow fragments as shown in Table 1 for this social media analysis task.

We first componentized the previously developed crawler and extractor as workflow fragments using the Wings framework and then re-used/re-purposed our workflow fragments to create the final workflow for human trafficking detection. The resulting workflow is shown in Figure 3 where the top black box labelled “Componentized Workflow Fragment” shows the original crawler and extractor incorporated as components in Wings. This is followed by:

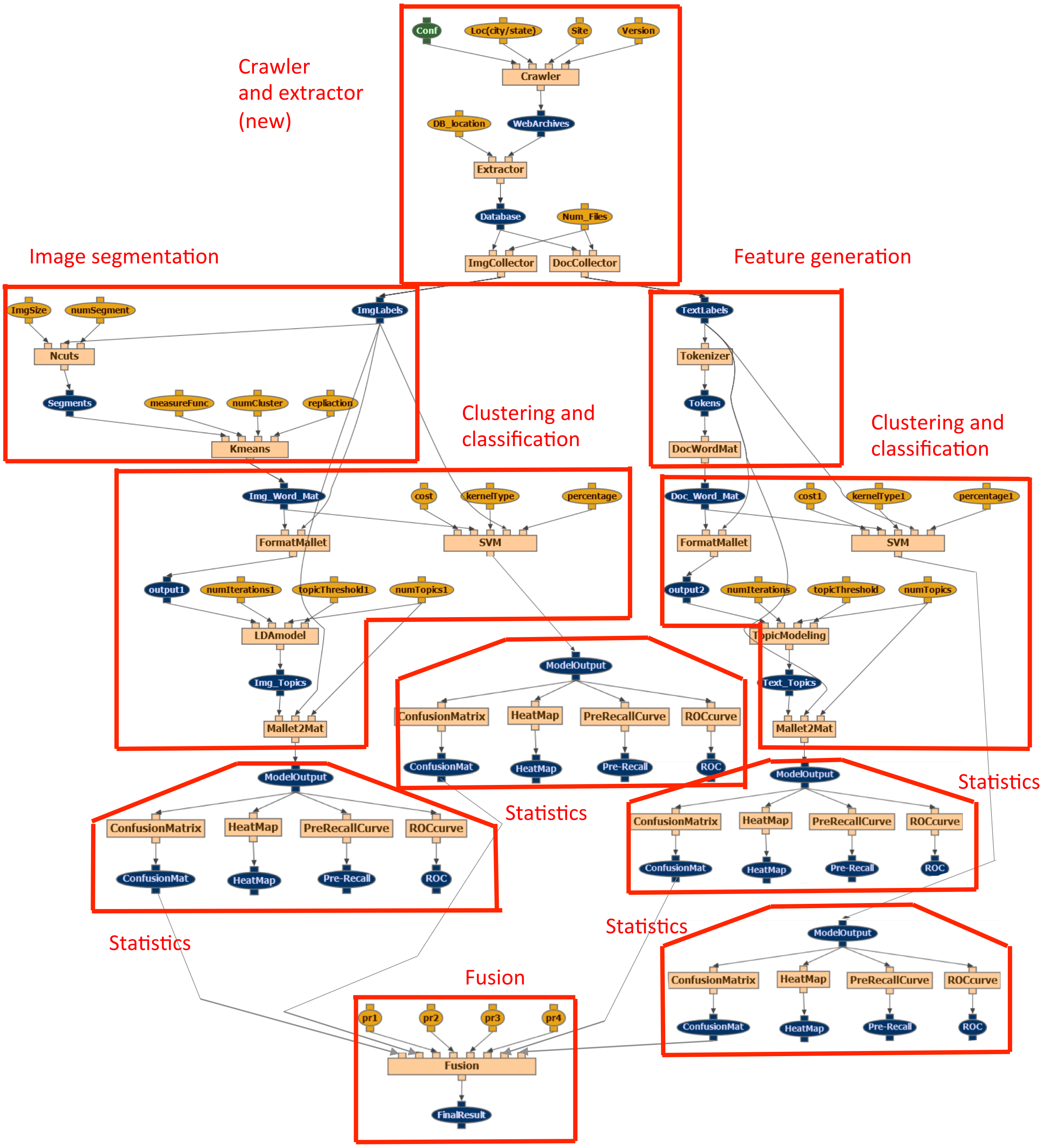


Figure 4: Detailed workflow for Fusion of Image and Text Analyses in the Multimedia Content Analysis workflow.

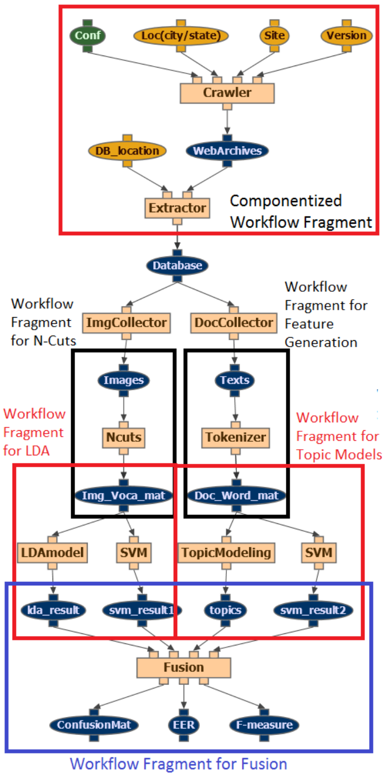


Figure 3: High level view of the workflow for Fusion of Image and Text Analyses in the Social Media Analysis workflow.

- Re-Use: The next two black boxes labelled "Workflow Fragment for N-Cuts" and "Workflow Fragment for Feature Generation" show re-use of the Image Analysis workflow fragments from Figure 2 as well as the re-use of the Text Analytics workflow fragments from Figure 1, respectively. Here, the "Tokenizer" component represents the entire workflow fragment in Figure 1(a).

- Re-Use: The next two red boxes labelled "Workflow Fragment for LDA" and "Workflow Fragment for Topic Models" show the re-use of workflow fragments for unsupervised analysis using MALLET and supervised analysis using SVM in a bag-of-words model from both the Image Analysis workflow fragments in Figure 2 and the Text Analytics workflow fragments in Figure 1. Here, the "TopicModeling" component represents the entire workflow fragment in Figure 2(d).

- Re-Purpose: The final blue box labelled "Workflow Fragment for Fusion" shows the re-purposed Fusion workflow fragment from Figure 2 for fusing the results of the Text and Image Analysis and visualizing those results.

Componentized	Crawler and Extractor Workflow Fragment
Re-Used	N-Cuts, Feature Generation, LDA, and SVM Workflow Fragments from Figures 1 [9] and 2 [11]
Re-Purposed	Fusion Workflow Fragment in Figure 2 [11]

Table 1: Workflow Fragment Componentization, Re-Use, and Re-Purposing.

	F-Measure	Equal Error Rate
Image-LDA	0.500000	0.523810
Image-SVM	0.526316	0.400000
Text-TopicModel	0.600000	0.428571
Text-SVM	0.470588	0.300000
Fusion	0.506283	0.488095

Table 3: Cumulative Statistical Comparison of all 5 models.

The final, detailed workflow is shown in Figure 4, where all the workflow fragments in Figure 3 are expanded. For the image analysis, an N-cuts algorithm is used. The N-cuts component takes two parameters: the image size and the number of segments. The image size simply means resizing the input image for the process. How many segments an image should be divided into is a user choice, so it varies depending upon the application. For our experiments, 10 segments were used. For each segment, a color histogram is computed and this information is used as the input for the K-means algorithm in the next step. Thus, a visual vocabulary based on the color histogram of each segment is built up. The number of clusters determines how large a visual vocabulary is created. For the distance measure in K-means, we used the Euclidean distance method. The centroid of each cluster determines the visual vocabulary word. Using a Bag-of-Words model, we can create an image-word document for all the image inputs using these visual vocabularies. This matrix is fed into the LDA and SVM algorithms.

6. EXPERIMENTS AND RESULTS

In this set of experiments, the goal is to determine if the person represented in an ad is a minor or not. The workflow uses both the ad text, as well as the ad images, to make this determination. We present below the results of our experiments and discuss the benefits of using workflows.

6.1 Exploring Different Parameter Values

After extracting images and texts separately from the original posts, each data element is analyzed using unsupervised and supervised learning algorithms. One benefit of using workflows is the ability to easily experiment with different parameter settings, while the system tracks the provenance of the results of each run.

For the unsupervised case of the image analysis, the Mallet LDA in our design takes three parameters: number of iterations, number of topics, and the topic threshold. The important parameter is topic threshold. If the topic proportion is less than the threshold decimal value, then this topic will not be displayed in the result. The topic threshold is often used as an efficient filter when there are many topics existing in the document set. For our case, since we're only making the determination of whether or not the subject of

(a)

	Predicted	
	Negative	**Positive**
Negative	0.250000	0.225000
Positive	0.275000	0.250000

(b)

	Predicted	
	Negative	**Positive**
Negative	0.250000	0.250000
Positive	0.200000	0.300000

(c)

	Predicted	
	Negative	**Positive**
Negative	0.300000	0.175000
Positive	0.225000	0.300000

(d)

	Predicted	
	Negative	**Positive**
Negative	0.200000	0.300000
Positive	0.150000	0.350000

(e)

	Predicted	
	Negative	**Positive**
Negative	0.250000	0.237500
Positive	0.250000	0.262500

Table 2: Confusion Matrices for: (a) Image-LDA, (b) Image-SVM, (c) Text-TopicModeling, (d) Text-SVM, and (e) Fusion.

a post is a minor, we only have a binomial case. Most of the percentages are close to 50%, either a little greater or a little less. Thus, setting topicThreshold to 0.1 ensures we get the topic in either case. After comparing the topic decision with the ground truth, the last step is to get the statistical result.

For the supervised learning case of the image analysis, we utilized an SVM. We did not want to set the cost control parameter for the SVM to be too large, which would cause an overfitting to the data. In general, the boundary becomes smoother with smaller cost. We thus set the cost parameter for the SVM to 0.5 and chose the linear kernel (kernelType = 0). The percentage parameter determines how to divide training and testing set and we set it to the simplest case of 50%. At the end, the SVM component outputs a comparison between the ground truth and expected label for each image, and this result is fed into the statistical module, as well.

For the text analysis side, the same components are used. The difference is in the processing of the data prior to getting a document-word matrix. Using a tokenizer, a unique word list is created from all text inputs. Then, this list is used to build a document-word matrix. With the matrix, the rest of workflow is the same as in the image analysis case above.

6.2 Managing Multiple Code Versions

The first steps of the workflow include a crawler that downloads the data from the desired site. Given a location, consisting of both the city and state, and the URL of the site where these ads are posted, the crawler component gathers all the web posts on that site. Different scripts need to be written to crawl different types of sites. The crawler component includes a parameter that allows the user to select a different version of the crawler code based on the desired script version. Thus, a benefit of using this workflow framework is that the system can manage multiple versions of codes.

6.3 Analysis of a Labeled Dataset

We have tested the workflow with a sample set of data which consisted of 40 ads, 20 of which were labelled for training and 20 were used for testing. One of the main hindrances to creating a larger test dataset is the dearth of hand-labelled data by law enforcement experts. We are currently labelling a larger dataset to run the analysis with more in-depth results.

The workflow itself took approximately three minutes to run after the data was collected, extracted, and labelled. We show confusion matrices for the Image-LDA, Image-SVM, Text-TopicModeling, Text-SVM, and Fusion methods in Table 2. We further show cumulative statistical measures of Equal Error Rate and the F-Measure for each of the five methods in Table 3. Here, we see the Text-TopicModel and Text-SVM methods tend to outperform since the Fusion module is a simple average scheme. In future work, we intend to experiment with various fusion methodologies in addition to this simple boosting scheme, including utilizing more complex learner combination schemes in the fusion step as we increase the number of methods utilized in the overall workflow.

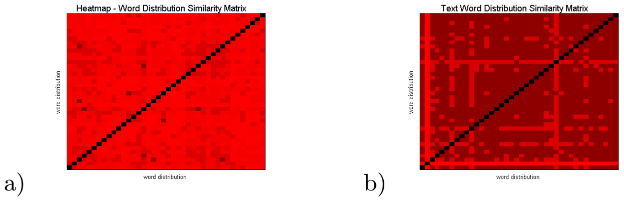


Figure 5: Heatmaps to identify duplicate posts in the large sample dataset using a) image analysis and b) text analysis.

6.4 Analysis of an Unlabeled Dataset

In addition, we used the workflow to examine a larger dataset of 6,000 posts that have not been labelled to find duplicate posts within that dataset; duplicate posts would indicate the same trafficked minor is transferred to multiple cities or locations. Results for the duplicate posts are shown as heatmaps for the image analysis and text analysis components in Figure 5. To use the workflow with unlabeled data, we simply weighted down the supervised learning results.

6.5 Experimenting with Different Workflow Fragments

Adding in more modules into the workflow easily is one of the greatest advantages of using scientific workflows for rapid application development. We can incorporate the latest age estimation computer vision methods by componentizing them in exactly the same way as we componentized the crawler and extractor. In addition, we can easily vary the parameters to run multiple experimental configurations. This capability to experiment by including or excluding certain modules and adjusting parameters is exactly the kind of ability the final product will enable law enforcement agents to utilize. The ability to change configuration and include different methods is shown in [9], where high school interns utilized workflow fragments to experiment with a wide variety of experimental setups and customizations.

(a)

	Predicted	
	Negative	**Positive**
Negative	0.250000	0.237500
Positive	0.212500	0.300000
F-Measure	0.524226	
EER	0.413095	
Proportions	imgLDA = 1, imgSVM = 1,txtLDA = 1,txtSVM = 1	

(b)

	Predicted	
	Negative	**Positive**
Negative	0.241667	0.250000
Positive	0.200000	0.308333
F-Measure	0.515635	
EER	0.392063	
Proportions	imgLDA = 0.5, imgSVM = 1, txtLDA = 0.5, txtSVM = 1	

(c)

	Predicted	
	Negative	**Positive**
Negative	0.258333	0.225000
Positive	0.225000	0.291667
F-Measure	0.532817	
EER	0.434127	
Proportions	imgLDA = 1, imgSVM = 0.5, txtLDA = 1, txtSVM = 0.5	

(d)

	Predicted	
	Negative	**Positive**
Negative	0.250000	0.237500
Positive	0.204167	0.308333
F-Measure	0.527915	
EER	0.396825	
Proportions	imgLDA = 0.5, imgSVM = 0.5, txtLDA = 1, txtSVM = 1	

Table 4: Different weighting schemes for the Modified Fusion module from Figure 5 showing confusion matrices and summary statistics for four different runs of the modified Fusion module with the following schemes: (a) equally weighted supervised and unsupervised methods, (b) greater weighting on the supervised module, (c) greater weighting on the unsupervised module, and (d) greater weight on the text modules.

Workflows allow us to prototype faster, include or exclude multiple methods, parallelize for big data use, scale up in terms of data and processing, and allow experimenting with various setups and different parameter values. This ability to componentize existing programs and expand the functionality by using pre-existing workflow fragments also results in significant time and effort savings. Here, we show how easy it is to vary the parameter values and do multiple runs to empirically determine the optimal fusion weighting. We first modified the Fusion component to allow the user to specify the weighting proportion for each input method, as shown in Figure 2 (f). We then experimented with varying the weights on the input modules by using equally weighted supervised and unsupervised methods (all modules are set to 1), greater weighting on the supervised module (LDA = 0.5, SVM = 1), greater weighting on the unsupervised module (LDA = 1, SVM = 0.5), and greater weight on the text modules (IMG = 0.5, TXT = 1). The results with confusion matrices and summary statistics for the four different runs of the modified Fusion module are shown in Table 4.

6.5.1 Analysis of Time/Work Savings

In this implementation, we incorporated the original crawler and extractor into Wings and then added on various Text Analysis and Image Analysis workflow fragments, including fusing their results and adding components to help visualize the results. This involved writing simple component wrapper scripts for both of the existing python scripts and setting up the mySQL database interface. The original development of the python version of the crawler/extractor had taken several months; this was quite involved as appropriate algorithms had to be researched, in addition to developing the code. The original crawler and extractor were componentized into Wings components, as shown in Figure 3.

This process took roughly two days as the original programs had to be made independent of the original development environment, account for supporting libraries, and had to interface with the external Database system that was distributed on the Web. Once this was done, the extension of the other components via workflow fragments for Image Analysis, Text Analysis, and their Fusion and visualization, took approximately one day, saving effort estimated to be on the order of 300 man-hours of work. This was estimated by the original developers using one postdoc and one graduate student working at a similar pace as in the development of the original prototype as they worked to identify appropriate algorithms for image and text analysis, implement them and incorporate them into their nascent crawler/extractor prototype, and then investigated a fusion methodology as well as the tools to visualize and analyze the results.

7. CONCLUSION

The multimodal information fusion of image and text data afforded by workflows helped to extend the Human Trafficking Detection (HTD) project in this preliminary work. In addition to extending the actual analysis, it also helped significantly reduce the time and work required in the development of those extensions. The benefits of workflows for mutlimedia content analysis for security applications, in fact, goes beyond this, as well: workflows allow tracking of intermediate results, mapping to various execution resources, failure recovery, and full load management [8]. In future work, we intend to show how the ability to parallelize data processing by executing workflows in distributed resources would allow us to address the volume and velocity of big data that presents a great challenging in the HTD application as well as other multimodal information fusion problems.

8. ACKNOWLEDGMENT

We gratefully acknowledge support from the Defense Advanced Research Projects Agency with award FA8750-13-C-0016. This research was also partially supported by the

National Science Foundation under Grant #1019343 to the Computing Research Association for the CIFellows Project. Thanks are due to our colleagues in federal law enforcement and the Los Angeles Metro Task Force on Human Trafficking.

9. REFERENCES

[1] S. C. Herring, "Web content analysis: Expanding the paradigm," in *International Handbook of Internet Research* (J. Hunsinger, L. Klastrup, and M. Allen, eds.), pp. 233–249, Springer Netherlands, 2010.

[2] C. Weare and W.-Y. Lin, "Content analysis of the world wide web: Opportunities and challenges," *Social Science Computer Review*, vol. 18, no. 3, pp. 272–292, 2000.

[3] M. Latonero, "Human trafficking online: The role of social networking sites and online classifieds," tech. rep., USC Annenberg School for Communication, 2011.

[4] T. M. Kurc, S. Hastings, V. S. Kumar, S. Langella, A. Sharma, T. Pan, S. Oster, D. Ervin, J. Permar, S. Narayanan, Y. Gil, E. Deelman, M. W. Hall, and J. H. Saltz, "High performance computing and grid computing for integrative biomedical research," *Journal of High Performance Computing Applications*, vol. 23, no. 3, pp. 252–264, 2009.

[5] A. Goderis, U. Sattler, P. Lord, and C. Goble, "Seven bottlenecks to workflow reuse and repurposing," in *The Semantic Web*, pp. 323–337, Springer Berlin / Heidelberg, 2005.

[6] Y. Gil, V. Ratnakar, E. Deelman, G. Mehta, and J. Kim, "Wings for pegasus: Creating large-scale scientific applications using semantic representations of computational workflows," in *AAAI*, pp. 1767–1774, 2007.

[7] Y. Gil, V. Ratnakar, J. Kim, P. A. Gonzalez-Calero, P. Groth, J. Moody, and E. Deelman, "Wings: Intelligent workflow-based design of computational experiments," *IEEE Intelligent Systems*, vol. 26, no. 1, 2011.

[8] Y. Gil, P. Szekely, S. Villamizar, T. Harmon, V. Ratnakar, S. Gupta, M. Muslea, F. Silva, and C. Knoblock, "Mind your metadata: Exploiting semantics for configuration, adaptation, and provenance in scientific workflows," in *Proceedings of the Tenth International Semantic Web Conference*, 2011.

[9] M. Hauder, Y. Gil, R. Sethi, Y. Liu, and H. Jo, "Making data analysis expertise broadly accessible through workflows," in *In Proceedings of the Sixth Workshop on Workflows in Support of Large-Scale Science (WORKS), held in conjunction with SC*, 2011.

[10] R. J. Sethi, H. Jo, and Y. Gil, "Re-using workflow fragments across multiple data domains," in *Supercomputing WORKS*, 2012.

[11] R. J. Sethi, H. Jo, and Y. Gil, "Structured analysis of the isi atomic pair actions dataset using workflows," *Pattern Recognition Letters*, vol. SI:SAHAR, 2013.

[12] I. Bhattacharya and L. Getoor, "Collective entity resolution in relational data," *ACM Trans. Knowl. Discov. Data*, vol. 1, Mar. 2007.

[13] J. Kim, Y. Gil, and M. Spraragen, "Principles for interactive acquisition and validation of workflows," *Journal of Experimental and Theoretical Artificial Intelligence*, 2009.

[14] I. J. Taylor, E. Deelman, D. B. Gannon, and M. Shields, *Workflows for e-Science: Scientific Workflows for Grids*. Springer-Verlag, 2006.

[15] Y. Gil, P. Szekely, S. Villamizar, T. C. Harmon, V. Ratnakar, S. Gupta, M. Muslea, F. Silva, and C. A. Knoblock, "Mind your metadata: exploiting semantics for configuration, adaptation, and provenance in scientific workflows," in *ISWC*, 2011.

[16] M. Hauder, Y. Gil, and Y. Liu, "A framework for efficient text analytics through automatic configuration and customization of scientific workflows," in *In Proceedings of the Seventh IEEE International Conference on e-Science*, (Stockholm, Sweden), 2011.

[17] I. H. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes, and S. J. Cunningham, "Weka: Practical machine learning tools and techniques with java implementations," 1999.

[18] B. Song, R. J. Sethi, and A. K. Roy-Chowdhury, "Robust Wide Area Tracking in Single and Multiple Views," in *Visual Analysis of Humans* (T. B. Moeslund, L. Sigal, V. Krüger, and A. Hilton, eds.), pp. 1–18, Springer-Verlag, 2011.

[19] N. M. Nayak, R. J. Sethi, B. Song, and A. K. Roy-Chowdhury, "Motion Pattern Analysis for Event and Behavior Recognition," in *Visual Analysis of Humans* (T. B. Moeslund, L. Sigal, V. Krüger, and A. Hilton, eds.), pp. 289–309, Springer-Verlag, 2011.

[20] R. J. Sethi and A. K. Roy-Chowdhury, "Modeling and Recognition of Complex Multi-Person Interactions in Video," in *ACM MM MPVA*, pp. 0–3, 2010.

[21] R. J. Sethi, A. K. Roy-Chowdhury, and S. Ali, "Activity Recognition by Integrating the Physics of Motion with a Neuromorphic Model of Perception," in *WMVC*, WMVC, 2009.

[22] R. J. Sethi and A. K. Roy-Chowdhury, "Physics-based Activity Modelling in Phase Space," in *ICVGIP*, 2010.

[23] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[24] A. K. McCallum, "Mallet: A machine learning for language toolkit," 2002.

[25] D. Garijo and Y. Gil, "A new approach for publishing workflows: Abstractions, standards, and linked data," in *SC WORKS*, (Seattle, Washington), 2011.

[26] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," pp. 412–420, Morgan Kaufmann Publishers, 1997.

[27] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *CSVT*, 2008.

[28] J. Aggarwal and M. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys*, 2012.

[29] M. Ryoo and J. Aggarwal, "Stochastic representation and recognition of high-level group activities," *CVPR*, 2009.

[30] J. Aggarwal and Q. Cai, "Human motion analysis: A review," *CVIU*, 1999.