Making Data Analysis Expertise Broadly Accessible through Workflows

Matheus Hauder
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
+1-310-822-1511
matheus@isi.edu

Yolanda Gil and Ricky Sethi
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
+1-310-822-1511
gil@isi.edu, rsethi@isi.edu

Yan Liu and Hyunjoon Jo
Computer Science Department
University of Southern California
941 Bloom Walk
Los Angeles, CA 90089-0781
+1-213-740-4371
yanliu.cs@usc.edu,
hyunjoon@usc.edu

ABSTRACT
The demand for advanced skills in data analysis spans many areas of science, computing, and business analytics. This paper discusses how non-expert users reuse workflows created by experts and representing complex data mining processes for text analytics. They include workflows for document classification, document clustering, and topic detection, all assembled from components available in well-known text analytics software libraries. The workflows expose to non-experts expert-level knowledge on how these individual components need to be combined with data preparation and feature selection steps to make the underlying statistical learning algorithms most effective. The framework allows non-experts to easily experiment with different combinations of data analysis processes, represented as workflows of computations that they can easily reconfigure. We report on our experiences to date on having users with limited data analytic knowledge and even basic programming skills to apply workflows to their data.

Categories and Subject Descriptors
C. Computer systems organization, D.2 Software engineering, D.2.10 Design.

General Terms
Design, Performance, Human Factors.

Keywords
Scientific workflows, text analytics, semantic workflows.

1. INTRODUCTION
In a world with increasingly more on-line information and with myriads of sensors, our ability to mine data is key to scientific discoveries, societal change, and business entrepreneurship. In science, vast amounts of data are collected in many disciplines and made openly available for analysis [1,2,11], whether virtual observatories in astronomy (http://www.sdss.org) or repositories of biomedical data (http://www.tcga.org). Data analytics has emerged as a widely desirable skill in many areas where discoveries are sought, from monitoring environmental cyberobservatories, to correlating on-line user behaviors, to aggregating medical records. Although foundational knowledge is taught in major universities and colleges, advanced data analytics can only be acquired through hands-on practical training. Only exposure to real-world datasets allows students to learn the importance of preparing and cleansing the data, designing appropriate features, and formulating the data mining task so that the data reveals phenomena of interest. However, the effort required to implement such complex multi-step data analysis systems and experiment with the tradeoffs of different algorithms and feature choices is daunting. For most practical domains, it can take weeks to months for a student to setup the basic infrastructure, and only those who have access to experts to point them to the right high-level design choices will endeavor on this type of learning. As a result, acquiring practical data analytics skills is out of reach for many students and professionals, posing severe limitations to our ability as a society to take advantage of our vast digital data resources.

We view workflows as a paradigm to: 1) expose non-experts to well-understood end-to-end data analysis processes that have proven successful in challenging domains and represent the state-of-the-art, and 2) allow non-experts to easily experiment with different combinations of data analysis processes, represented as workflows of computations that they can easily reconfigure and that the underlying system can easily manage and execute.

This paper describes a highly reusable family of workflows for text analytics, which includes workflows for document classification, document clustering, and topic detection. These workflows capture expertise on using supervised and unsupervised statistical learning algorithms, as they reflect state-of-the art methods to prepare data, extract features, down-select features, and train models of the data. Our framework uses the Wings workflow system [8], which has two key features that make workflows accessible to users with limited programming skills: a simple dataflow structure and a simple web interface. This paper also reports on our experiences to date having non-expert users apply these text analytics workflows to their data and extend them to suit their analytic tasks.

Work to date on workflow reuse has focused on expert scientists reusing workflows from other scientists [4], and to our knowledge our work is the first to look at reuse of expert workflows by non-experts. While reuse by other expert scientists saves them time and effort, reuse by non-experts is an enabling matter as in practice they would not be able to carry out the analytic tasks without the help of workflows.

WORKS'11, November 14, 2011, Seattle, Washington, USA.Copyright 2010 ACM 978-1-4503-1100-7/11/11…$10.00.

We begin motivating the value of workflows as a means to represent valuable expertise in data analytics, and in particular text analytics. We describe our approach, describing the workflows that we have implemented for text analytics, how they are represented in Wings, and how users interact with the system to browse and run workflows. We then present our experiences to date with non-experts reusing text analytics workflows. We finalize with conclusions and thoughts for future work.

2. MOTIVATION

Data analytics skills cannot easily be acquired from books or in a classroom setting. Many courses on different levels of statistics, machine learning and data mining are offered in most universities and train students on the relative merits of different algorithms and statistical techniques. However, in practice designing an appropriate end-to-end process to prepare and analyze the data plays a much more influential role than using a novel classifier or statistical model. For example, for text analytics, in many cases the prediction accuracy of text classifiers on one dataset can differ 5-10% depending on how the unstructured texts are converted to feature vectors. In contrast, once the data is preprocessed the difference between which classifier (such as support vector machines or Naïve Bayes) is applied on the same feature vector is only 0.5-5%. Moreover, state-of-the-art data analytics often involves multi-step methods with sophisticated statistical techniques such as cross-validation and combinations of algorithms such as ensemble methods. Such methods are challenging to set up and run and few users will have requisite experience and infrastructure to experiment with them. Finally, such expertise can only be learned by experiencing the performance of different methods with real data, by understanding different data preparation strategies, and by exploring the relative merits of different algorithmic choices and their effect in the overall performance.

In research projects and industrial practice, advanced data analytic skills are usually achieved by working on multiple data analytic task domains and being coached by experts. This places a significant barrier for researchers that are interested in acquiring these skills but do not have access to such settings. First, developing an appropriate setup for any real problem (e.g. email prioritization) requires a good understanding of the state-of-the art analytics in that domain (e.g., text analytics), placing a barrier for many students who do not have easy access to that expertise. Moreover, significant software infrastructure needs to be deployed in order to learn by applying and observing different techniques in a real problem domain, requiring significant investment which deters potential students from attempting to do so. Setting up this infrastructure requires programming skills, making it infeasible for students without significant computer science background. Finally, the cycle to develop appropriate infrastructure in a real domain can be as long as months or years, making it impractical for students who want to acquire these skills to do it in one and much less in several domains.

The use of workflows for representing and managing complex scientific data analysis processes has been described in [2,11,6,17]. Workflows represent complex applications as a dependency network of individual computations linked through control or data flow. Workflows effectively capture valuable expertise, as they represent how an expert has designed computational steps and combined them into an end-to-end process.

Our target users include students, researchers and practitioners who intend to use data analytics in industry or scientific research. A pre-requisite to use our system is to take necessary courses and training materials to be familiar with basic machine learning and statistical data analysis techniques. The goal of our work is to supplement that material with practical learning experiences.

Our goal is to significantly reduce the learning cycle since the students can utilize existing components to work on different workflows and setup experimental runs within minutes, while the usual cycle of implementing a process from raw input to final results is on the scale of months or years, even given that some components can be downloaded from shared sources. For example, in many computational biology applications, it may take an inexperienced student several months to implement a basic protein secondary structure prediction framework that consists of sequence analysis, feature extraction, classifiers, and postprocessing with decent performance. In our system, a student will be able to achieve this in several minutes. Our system provides an effective solution to lower the barriers to learning advanced skills for data analytics.

Our work will also enable access to data analytics training experiences for students who have no computer science or programming background. For example, many students in statistics or bioinformatics end up being limited to painstakingly reformatting and preparing data by hand or using only what is available in end-user environments such as MATLAB. Our system will provide real-world datasets and an extensive list of already-packaged state-of-art data analysis components, such as feature extraction, feature selection, classifiers, unsupervised learning algorithms, and visualization tools. It will enable non-programmers to experiment with this rich set of components by easily assembling them into end-to-end data analytic processes represented as workflows.

Our work will also target researchers that have developed initial data mining applications and are seeking to improve the performance of their application. A good example here is compiler optimization, where the use of data mining techniques is being adopted in order to rapidly customize optimizations to new computer architectures that come out every few months. In carrying out a recent survey of this research area, we found that most of the work focuses on older techniques that are far from the state-of-the-art in data analytics [9]. Lowering the cost of learning data analytics skills would enable compiler researchers to achieve new levels of performance. Similarly, sophisticated analytic skills are required to analyze the reams of data in mobile devices and other human-computer interfaces.

Finally, expert-level data analytics practitioners would also be users of our system to learn new techniques. Experts can read and be aware of the newest algorithms, but currently do not have a practical means to obtain hands-on experience with them because they require a large investment of effort. Moreover, experts often reach a comfort zone with algorithms and techniques that they have experience with, and are reluctant to invest the effort to learn novel state-of-the-art methods. We envision sustaining learning as a long-term activity throughout a professional career, so that experts can keep up with research innovations in an easier, time-efficient, and hands-on manner.

3. WORKFLOWS FOR TEXT ANALYTICS IN WINGS

We use the Wings workflow system [8] equipped with several expert-quality workflows that represent a powerful set of text analytic methods [10]. The framework includes workflows for tasks such as document classification, document clustering, and topic modeling. These workflows are composed of workflow fragments that pre-process text, prepare the data, and set up the learning task. The workflows are composed of more than fifty workflow components that we built using popular machine learning and text processing packages, including Weka [18], CLUTO [12], and MALLET [14] among others. These packages have very heterogeneous implementations but the components encapsulate the software with interfaces described with data types in the workflow system to make them reusable in different workflows. The workflow system 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 community. A technical overview of the framework is described in [10], given from a developer's perspective. In this section, we focus on the user's perspective, illustrating the collection of workflows that the framework provides.

3.1 Workflow Fragments

The workflows are composed of workflow fragments that are reused across workflows. These predefined workflow fragments make text analytics expertise readily available to new users.

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). In the first step common stop words (e.g., a, for, the) are removed from the data set since they don't improve the learning performance. Next to the stop words another component also removes words that are smaller than a given size. This will also remove special characters from the dataset. Morphological variations are removed from the dataset with a stemmer component. The stemmer component is especially denoted with a dashed box because it is an example for an abstract component. These components have further specialized components and represent possible variation steps in the workflow. For this particular component the framework can choose between a Porter Stemmer and a Lovins Stemmer. The last step for this workflow is the term weighting that is used to transform the dataset into the vector space model format. Since this is an abstract component one can choose between various different implementations. 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.

Feature Selection: A very common step for many classification problems is the feature selection shown in Figure 1(b). Main purpose of the feature selection 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 the correlation score. Typical implementations for this step are Chi Squared, Mutual Information or Information Gain that can be found in [19] 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 efficiently handle large matrices, the component for the feature selection uses an implementation written in Java.

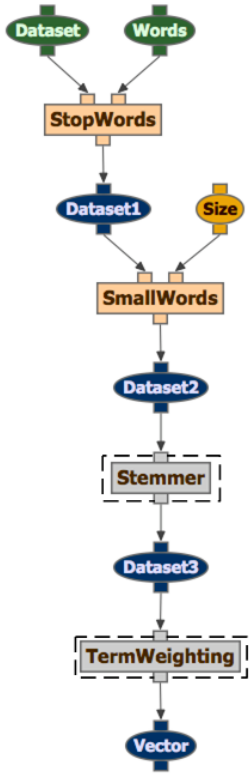
Training and Classification: The resulting training set after the feature selection can be used for the training of a model with the workflow shown in Figure 1(c). Both components in the workflow use the Weka machine learning framework. Thus, many different machine learning algorithms can be used to 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 used components 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.

A constraint in the component catalog of the framework also makes sure that the workflow is executed correctly. In a workflow that is not executed correctly one could use different machine learning algorithms for the modeler and the classifier components. In the framework however the user doesn't have to deal with this potential problem, because the invalid instances are automatically rejected by the workflow system and is not possible to execute them.

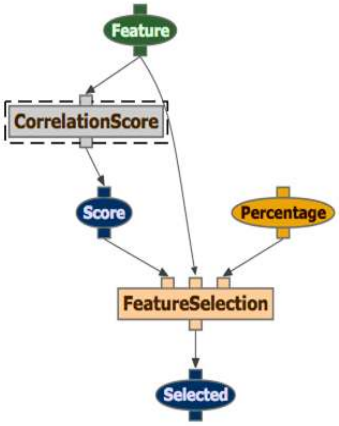
Clustering: The workflow fragment for clustering is shown in Figure 1(d). The Vector that results from the Feature Generation workflow can be used as input for clustering. It needs to be formatted into the suitable format for the clustering software. The result of this step is the Feature output with 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.

3.2 End-to-End Workflows

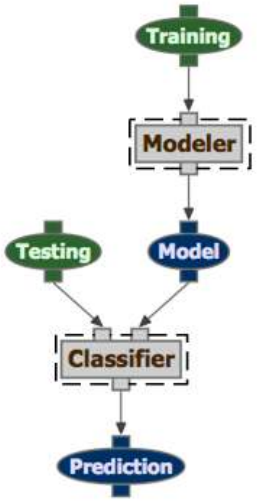
The previously defined workflow fragments can be executed independently from each other. Some researchers might focus on some particular parts in order to optimize or improve their understanding of the behavior in the individual steps. A good starting point for novice researchers however is to use end-to-end workflows that are formed using the components and workflow fragments discussed above. These end-to-end workflows represent advanced expertise in that they capture complex combinations of components that are known to work well in practice. These workflows are pre-defined by experts and available as part of the workflow library. They can be executed with available datasets, or adapted by adding or changing components.



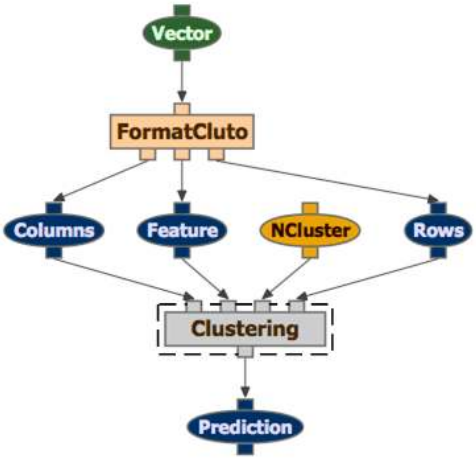
(a)



(b)



(c)



(d)

Figure 1. Workflow fragments for (a) Feature generation, (b) Feature selection with correlation scoring, (c) Training a model to classify a test dataset, (d) Clustering of documents with label information. They are composed of common workflow components, for example in (a): "StopWords" removes common stop words (e.g., a, for, the) from the data set in "Words"; "SmallWords" removes words that are smaller than a given size determined by "Size"; "Stemmer" converts morphological variations from the dataset with a stemmer component; and "TermWeighting" transforms the dataset into the vector space model format.

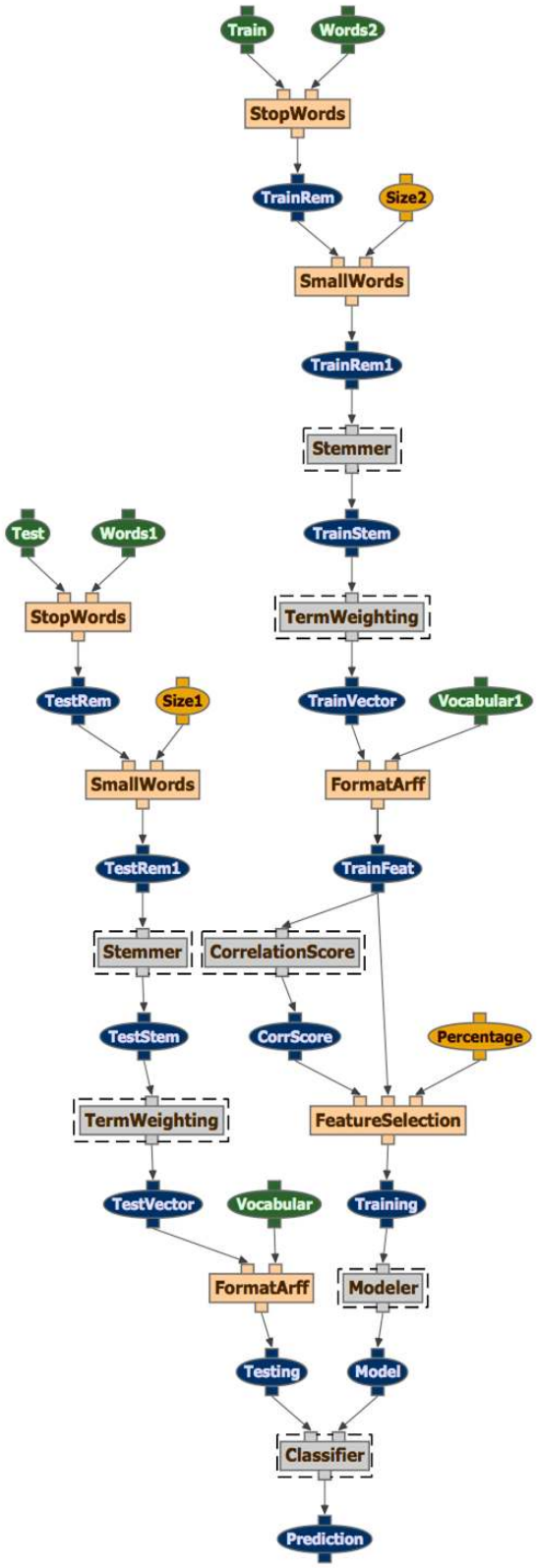


Figure 2. Workflow for document classification using a testing and training set.

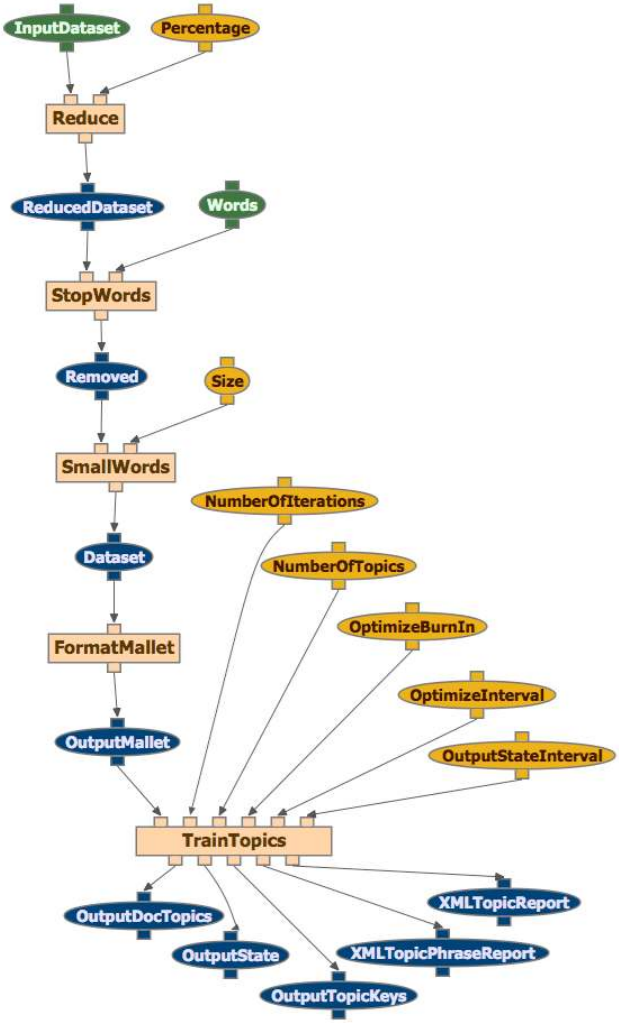


Figure 3. Workflow for topic modeling.

Document Classification: Figure 2 shows a document classification workflow. It has two branches for the testing and training set. For the branch with the training set, the feature selection workflow fragment is applied. There is an additional step in each branch to transform the data format so it can be processed by the modeler and classifier which is implemented with the Weka toolkit. For this purpose a Vocabulary file contains all words of the entire dataset to make sure that every word is mapped to a unique number. The final steps perform the modeling and classification of the datasets to achieve the predictions.

Document Clustering: The document clustering workflow is similar to the document classification workflow shown in Figure 2, except for the last steps where the modeler and classifier fragment that was shown in Figure 1(c) is replaced by the clustering workflow fragment in Figure 1(d).

Topic Modeling: The topic modeling workflow is shown in Figure 3. There are actually several workflows for topic modeling with small variations, for example some have stemming steps and some do not.

3.3 Experimenting with Workflows

A user would select a workflow, a dataset, and provide parameter values to run the workflow. Based on the dataset, the configurable parameters, and the number of abstract components, a list of executable workflows is generated. Since abstract components should be specialized within a workflow procedure, the Wings workflow system considers all their possible combinations by using a brute force approach. The Wings semantic reasoner, a part of the workflow system, automatically rejects invalid combinations of components if they violate the constraint rules specified in the data flow. In the document classification (Figure 2), for instance, the vector output of preprocessing should be converted to Arff file format. The FormatArff component is utilized for this task, and it takes two inputs; a vector output from Feature Generation and Vocabular. The Vocabular is the stemmed dataset using either Porter stemmer or Lovins stemmer (prior to this selection, we only see the Abstract Stemmer component). In this case, the selection of Vocabular processed by Porter stemmer will never consider the Lovins stemmer at the previous step (where we only see the Abstract Stemmer component). The two inputs provided to the FormatArff component should be compatible with each other, so the system knows it need not explore the possibility of choosing the other stemmer. As a result, the Wings workflow system prunes invalid branches. Hence, it saves time on experimental setups by only providing a correct execution table and frees any concerns of mis-configuration from a user. A detailed description of the user interface and the interaction dialogue is described in [Gil et al 2010].

4. RELATED WORK

In [16], prominent members of the machine learning community argue for the need to share software and datasets to facilitate experimentation and learning. There are already widely-used libraries such as MLC++ [13] and Weka [18]. The popularity of these systems demonstrates the demand for accessible machine learning codes. Although Weka provides basic functionality to compose codes into workflows, it does not provide any facilities to guide non-experts in how to combine them or how to prepare their data or select features that are appropriate for their goals.

Gestalt [15] is a user-centered environment for machine learning that is designed for programmers and guides them through pipelines that include data cleansing and visualization steps. However, it focuses only on classification tasks which are the simplest ones.

A workflow approach for text analytics is used in the IBM UIMA system [5], but it requires manual construction of the workflow including the interfaces between different components.

5. REUSE OF TEXT ANALYTICS WORKFLOWS BY NON-EXPERTS

To investigate the usability of our framework, we report on two very different cases of reuse of our workflows. The first is a case of reuse by researchers not expert in machine learning or text analytics, using the workflows for a project that targeted the analysis of a text corpus to improve a question/answering web site. The second is a case of reuse by high-school students for an internship project to analyze twitter data.

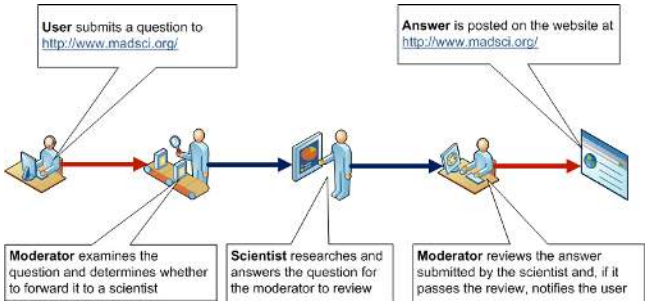


Figure 4. Overview of the question processing flow on The MadSci Network.

5.1 Reuse by Other Researchers

The Madsci Network is an Ask-A-Scientist website[1]. It provides a human-mediated Question & Answering (Q&A) service that answers questions in 26 different scientific fields. Boasting a store of over 40,000 questions and answers, it serves as a unique repository of scientific knowledge. However, with more than 650,000 unique visitors and only 700 scientists to answer questions, it is worth automating some of the processes that are currently done manually to handle user questions.

5.1.1 Research Questions for Q&A Framework

When a user first comes upon the site, they might immediately try to submit a question they have wrestled with for a while, as shown in the question process flow in Figure 4.

One of the issues with the operation of such an immense knowledge base is that it is difficult to automatically determine whether a new question has already been answered on the website. If it has not, the question is routed by the moderator to a scientist who is best suited to answer that query. However, finding scientists that are especially appropriate for a specific question is equally challenging given the vast number of scientists actively answering questions on the site. Finally, determining the correct category into which a question falls is another substantial machine learning task associated with Q&A sites as users often mis-categorize their queries. Thus, the main questions associated with analysis of The Madsci Network corpus are:

1. Automatic Question Answering: suggesting best matches from the archives for an incoming question
2. Task Assignment/Expert Finding: finding the best-suited scientist for incoming questions
3. Label Assignment: finding the most appropriate category for incoming questions

Concomitant with these research thrusts are several other issues, including dealing with short documents (e.g., the lengths of submitted questions,) and examining trends in the data that have applicability well beyond the specific corpus studied. A promising new approach to help address all of these data analysis problems is based on topic modeling.

Topic models [3] are a Bayesian graphical model-based approach to discovering hidden semantic topics in a corpus. One of the most popular tools which implements Latent Dirichlet Allocation, and its many variations, is MALLET, which is used in the Wings topic modeling workflow.

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

Just as with other machine learning methodologies applied to a specific corpus, topic models require in-depth and varied experimentation. Once the theoretical models have been established, significant experimentation is needed to determine model selection and parameter optimization, output analysis, and extensive evaluation of results for various experimental scenarios. This is especially important in topic modeling as no formal, structured approach to evaluation currently exists. Once the initial analysis and baseline is established, new models can be implemented and compared to the baselines.

5.1.2 Experimentation without Workflows

Non-Workflow analyses involve writing disparate scripts and software and keeping track of multiple experiments separately. This approach requires considerable expertise and is rife with experimental intricacies, especially of the implementation details as well as experimental provenance, where the experimenters have to keep track of the various parameters employed for each set of experiments. In the specific case of The Madsci Network, this involved:

- Experimentation with multiple approaches to pre-processing
- Learning intricacies of the MALLET software system
- Experimenting with various parameters of MALLET
- Evaluating the sizeable and plain text-only output of MALLET
- Implementing new models within the MALLET framework and repeating above experiments

The most difficult part of conducting a traditional empirical analysis, even for experienced researchers, is the enormous effort and specialized knowledge required to understand and setup the software and to keep track of the various approaches that were examined. The initial experimentation, in fact, involved two faculty, one postdoc, and one graduate student and required three months of concerted effort. Even then, the sheer administrative burden of evaluating and keeping track of the multitude of experiments proved onerous.

Using the non-workflow based experimentation, topic modeling and summary statistics were readily generated; however, they did not include visualization of trends and evaluation. Although it was possible do both, this was prohibitive due to the varied file formats and the sheer amount of work involved while continuing the initial topic modeling experimentation. In addition, some techniques like term weighting were not included in the initial experimentation which would have helped make the analysis more precise. These visualization steps and text processing techniques were available as part of the workflow framework, which made a difference as we describe next.

5.1.3 Experimentation with Workflows

In general, applying machine learning theories effectively and efficiently to real-world corpora requires extensive trial and error when dealing with the practical issues of model selection and optimization. This is where the Wings workflow system made the analysis much simpler, quicker, and complete. Using the Wings workflow system not only allowed easy specification of different components but also kept provenance information for each experiment, allowed insertion of multiple visualization and evaluation components, and enabled straight-forward customization/modification of existing experiments to include the incorporation of new models as they were developed.

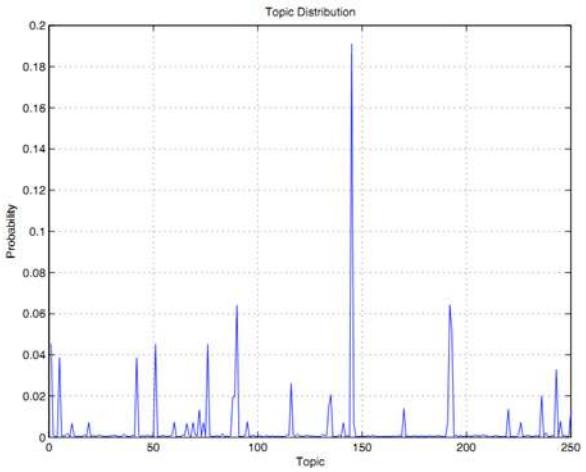


Figure 5: Topic distribution for a sample question.

We used the Wings workflow for topic modeling shown in Figure 3. The various parameters associated with MALLET, as well as the various outputs, can all be easily specified, customized, and used in subsequent processing, as shown below.

For example, we can easily take one of the MALLET outputs, the OutputDocTopics, which shows the distributions over topics for each document, and insert a Weka component to visualize it. This visualization is shown in Figure 5.

This is the plot of a single question, and its distribution over topics, which clearly shows the dominance of a single topic in the distribution. Such plots intuitively reveal insights about the individual questions and about the overall dataset.

In addition, this kind of visualization would easily allow comparison of the histograms of similar questions in order to determine the most similar questions and answers using simple distance measures which are inserted as components in the additional processing of the MALLET output. Initially, we got results for that experiment that had many category labels. Later, we used coarser-grained category labels for each document where the coarser grain categories are super-sets of the original labels. Table 1 shows the confusion matrix for the new categories. The initial results as well as examples of workflow variations created can be found in the project website[2].

Table 1: Confusion matrix for coarse-grained categories

	a	b	c	d	e	f	g
a	49	4	19	1	3	34	2
b	3	46	28	3	3	42	2
c	7	23	390	34	7	41	5
d	8	6	56	93	7	19	3
e	2	5	11	0	9	9	2
f	27	21	42	11	13	478	3
g	4	4	9	2	1	8	5

[2] http://workflow-sharing.isi.edu/workflow-sharing/index.php/Workflow_Reuse

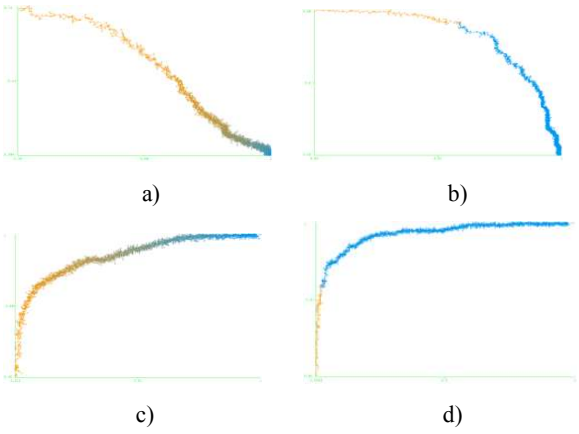


Figure 6: a) Precision-Recall curve for 5%; b) Precision-Recall curve for 15%; c) ROC curve for 5%; d) ROC curve for 15%.

In addition to visualizing the output in various ways, it is possible to vary the input dataset by easily adding a component to select only a subset. For instance, one experiment might specify the use of only 5% of the dataset in order to identify trends in the subset alone, as well as comparing with a larger proportion or even the entire dataset. This is exactly what is seen in 6a and 6c, where we see the Precision-Recall and the ROC curve, respectively, for the comparison of a single input question to all other questions and answers in that dataset in an effort to analyze research questions #1 and #2.

Figure 6b and Figure 6d show the same analysis for a larger component of the dataset (15%), also showing how the accuracy changes when the size of the dataset is varied. The only change required in the Wings framework was changing a single parameter on the input dataset as this is facilitated by a Reduce component to create datasets of varying percentages of the original size.

It is also relatively simple to analyze how questions and answers cluster together, using the clustering workflow in Figure 3. The results are shown in Figure 7. We can use this workflow to show how documents and topics cluster; this can be used by both the users and the moderator. When a new question is submitted, a new clustering diagram will be produced in which topics would be on the y-axis and documents on the x-axis; this would clearly show which questions/answers cluster with the correct answer (on the x-axis) for the user and which topics cluster together on the y-axis for the moderator to see which topics are most relevant for the new question.

The main realization was that using the Wings workflow system simplified the process of analysis significantly. It not only allowed calculation of standard statistics but also facilitated plotting of document-topics to help visualize it using CLUTO, allowed extensions of the MALLET toolkit (e.g., the Poly-Lingual Topic Model, PLTM, as well as new custom models) to be incorporated easily, with just as trivial replication of previous experimentation, allowed post-processing and visualizing of complex text output as shown in the Precision-Recall and ROC curves, as well as the histogram spectra of topic distributions, using tools like Weka.

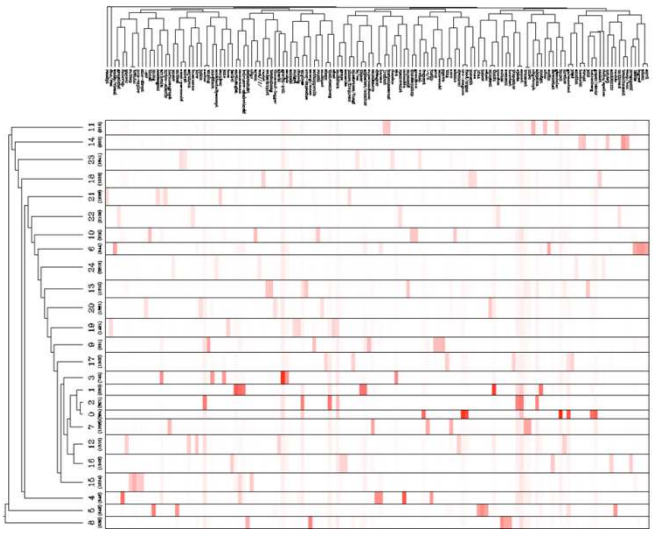


Figure 7: Clustering output for The Madsci Network dataset.

The web interface of the workflow system makes setting up an experiment very easy. The researchers could adapt the pre-existing workflows and make modifications according to the end application. Wings reasons about the workflows specified and ensures that they contain valid combinations of components. In the Feature Selection workflow, for instance, the selection of a Vocabular dataset that was built using the Porter stemmer will make the system reject the Lovins Stemmer component at the previous step because the same stemmer should be used to process the corpus and to create the Vocabular input.

Intermediate data results are accessible while executing the workflow, and this provides significant advantages for researchers and students. Therefore, they can compare which algorithm is the best fit to the given dataset for analysis. For example support vector machines (SVMs) are a very popular method for classification. The framework allowed one of the students involved to carry out several experiments to compared SVMs to other machine learning algorithms. In the experiments, the SVM classifier made better predictions on the WebKB_test dataset and the MadSci Vocabular dataset than Naïve Bayes or K-Nearest Neighbor algorithms. The Naïve Bayes classifier indicated the most erroneous predictions. Even if the classifier was 100% sure for some instances of the trained model, those probabilities were still associated with the wrong prediction. Through the experiments, the student was able to apply the acquired academic knowledge to a real-world application, and helped clarify how to make a choice for both the Modeler and Classifier steps. In the end, this experience allowed the student to acquire practical skills in advanced text analytics.

Finally, it was easy to extend the analysis to include alternate analytical methods, including replacing the topic models with word frequencies and repeating all of the previous experimentation for the new component.

To summarize, the main advantages realized using the Workflow-based system for these researchers were:

- Storing provenance information for tracking experimental protocols and results
- Using pre-existing components and working with a wide variety of pre-defined file formats

- Allowing simple plug-and-play of complex components that are prohibitive from a resource or time perspective without workflows
- Easy exploration of parameters for model selection/optimization
- Ability to customize components and design additional components

5.2 Reuse by High School Students

We recruited three high school students with limited programming background to use our system over a period of a week. The students had taken two semesters of introduction to programming in the eight and ninth grades, and were entering tenth grade in the coming year. After a short tutorial, they were then were asked to formulate useful tasks for themselves that would require running workflows or extending them by adding new components.

During the five days, the students did the following tasks:

- Became familiar with workflows as a software paradigm
- Learned to use the system and run simple workflows to analyze data (e.g., compare sets of html files to see how they would be classified)
- Learned to use pre-existing workflows for advanced text analytics (e.g., run workflows for document clustering and topic detection and compare their performance for different threshold parameters)
- Extended existing workflows with new workflow components that they developed
- Analyzed twitter data to detect topic trends by applying pre-existing advanced text analytic workflows

They also wrote a report describing these activities and their findings[3]. We highlight here two interesting accomplishments of their work.

5.2.1 Using Workflows to Learn to Select Features for Learning

The text classification workflow has a parameter "Percentage", which determines how many of all the possible features should be used by the learning algorithm. They wondered what was the right value to use. To answer that question, they run the workflow using Naïve Bayes as the learning method and using different values of that parameter. They reported the following observation:

> *"The graph demonstrates that at around 30% of included instances, the percentage of correctly classified instances levels for the data set we used."*

The curve is different for different machine learning algorithms, which they were able to explore as well by selecting different configurations of the workflow. This is a well-known phenomenon for machine learning experts, because the optimal number of features to use depends on the machine learning algorithm and the dataset at hand. With the workflow framework, the students were able to learn this easily by experimenting with the workflows.

[3]http://workflow-sharing.isi.edu/workflow-sharing/index.php/Workflow_Usability

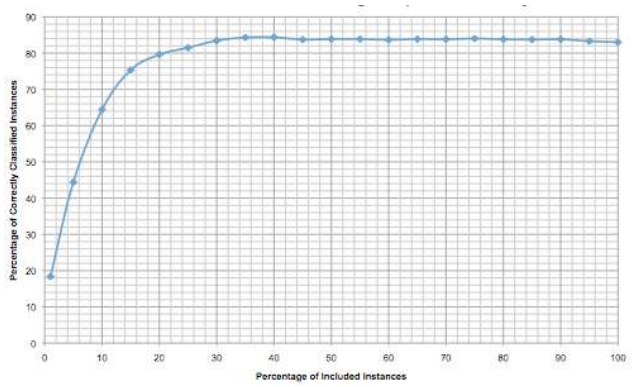


Figure 6: Plot showing how the parameter that selects the percentage of instances used for training affects classification accuracy.

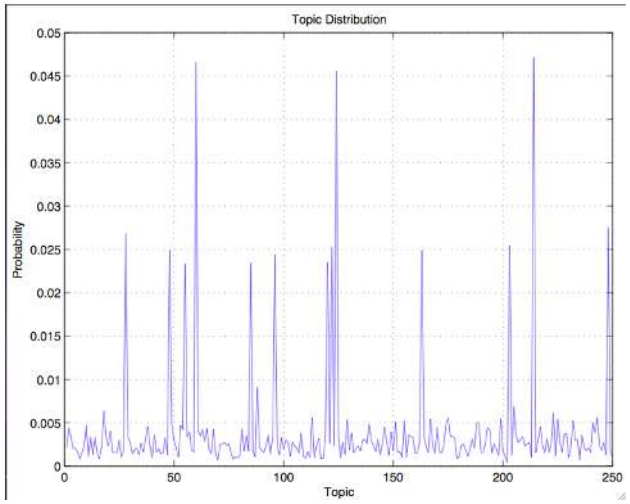


Figure 7: Plot of the highest ranked topics for one of the tweets.

5.2.2 Using Workflows to Analyze Twitter Data

We gave them a dataset that we extracted from the twitter.com site. The dataset has more than 250,000 tweets taken from November 1 2009 to February 28 2010, which includes the date when the Haiti earthquake occurred. They were curious about the most popular topics that were discussed in the dataset.

The data from twitter was in a format that was not appropriate for the workflows. Therefore, the students had to pre-process the data, and for that they wrote three pre-processing components that they executed before running the workflow. First, the data included tweets in several languages. To address this, they wrote a component that selected tweets that were only in English, by looking for common English words such as "the" and "and". Second, the dataset had html markup tags, which would result in confusing features for the machine learning algorithms. To address this, they created a component that finds and removes html tags. Finally, the dataset was also full of strings of non-alphabetic characters, such as URLs, and those would not be appropriate features. They wrote a third component to extract only the words formed by alphabetic characters.

Once the data were formatted appropriately with the new components that they wrote, they run the topic modeling workflow. With a simple keystroke, they were unknowingly using state-of-the-art methods for this task, and were able to generate Figure 7 showing the highest ranked topics for one of the tweets. Other plots could be created to show the most popular topics of the entire dataset.

6. CONCLUSIONS

The demand for advanced skills in data analytics spans many classic and emerging domains, including social network analysis, bioinformatics, cybersecurity, climate science, and business analytics to name a few. We have shown in this paper a framework based on scientific workflows that has been used to capture expertise from domain experts in data mining and text analytics. Our preliminary results show that the framework can be used by non-experts to carry out sophisticated data analysis tasks, even when they have very limited programming skills. Non-experts can reuse and extend these workflows to customize them for new data and new applications. An important capability that is missing from our system is to create plots and visualizations that aggregate results from several workflow runs. Our users did this by hand, and it would be nice if the system had a notion of workflow collections and allowed users to create visualizations of results along selected dimensions.

Our work is a step towards a framework that could make data analytics accessible to students, scientists, and professionals who lack the programming skills required to assemble themselves end-to-end data analysis systems for experimentation and practical learning. If we succeed, the wide adoption of our approach could ultimately lead to broad societal impact by changing the way people interact with data, learn from using scientific data, and participate in scientific data analysis tasks.

7. ACKNOWLEDGMENTS

This research was supported in part by the US National Science Foundation (NSF) with grant number IIS-0948429, in part under grant #1019343 to the Computing Research Association for the CIFellows Project.

8. REFERENCES

[1] "Petascale Computational Systems: Balanced Cyber-Infrastructure in a Data-Centric World ," Gordon Bell, Jim Gray, Alex Szalay, Letter to NSF Cyberinfrastructure Directorate, IEEE Computer, 39(1), pp 110-112, January, 2006.

[2] "Beyond the Data Deluge," Gordon Bell, Tony Hey, and Alex Szalay. Science, 6 March 2009.

[3] "Latent Dirichlet Allocation." Blei, D. M., Ng, A. Y., & Jordan, M. I. *Journal of Machine Learning Research*, 993–1022, 2003.

[4] "The design and realizations of the myExperiment Virtual Research Environment for social sharing of workflows". De Roure, D; Goble, C.;Stevens, R. Future Generation Computer Systems, 25 (561-567), 2009.

[5] "UIMA: an architectural approach to unstructured information processing in the corporate research environment." David Ferrucci and Adam Lally. 2004. Nat. Lang. Eng. 10, 3-4, 327-348.

[6] "Examining the Challenges of Scientific Workflows." Gil, Y.; Deelman, E.; Ellisman, M. H.; Fahringer, T.; Fox, G.; Gannon, D.; Goble, C. A.; Livny, M.; Moreau, L.; and Myers, J. 2007. IEEE Computer, 40(12):24-32.

[7] "Assisting Scientists with Complex Data Analysis Tasks through Semantic Workflows." Yolanda Gil, Varun Ratnakar, and Christian Fritz. Proceedings of the AAAI Fall Symposium on Proactive Agent Assistants, November 2010.

[8] "Principles for Interactive Acquisition and Validation of Workflows." Gil, Y., Kim, J., and Spraragen, M. To appear in the Journal of Experimental and Theoretical Artificial Intelligence, 2011.

[9] "Self-Configuring Applications for Heterogeneous Systems: Program Composition and Optimization Using Cognitive Techniques. " Hall, M.; Gil, Y.; and Lucas, R. 2008. In Proceedings of the IEEE, Special Issue on Cutting-Edge Computing: Using New Commodity Architectures, Volume 96.

[10] "A Framework for Efficient Text Analytics through Automatic Configuration and Customization of Scientific Workflows", Matheus Hauder, Yolanda Gil, and Yan Liu. Proceedings of the Seventh IEEE International Conference on e-Science, Stockholm, Sweden, December 5-8, 2011.

[11] "The Fourth Paradigm: Data-Intensive Scientific Discovery" Hey T., Stewart Tansley, S., and K. Tolle. Microsoft Research, 2009.

[12] "CLUTO a clustering toolkit," G. Karypis, Department of Computer Science, University of Minnesota, Technical Report 02-017, 2002. Available from http://www.cs.umn.edu/~cluto.

[13] "Data Mining Using MLC++: A Machine Learning Library in C++. Tools with Artificial Intelligence," R. Kohavi, D. Sommerfield, and J. Dougherty, IEEE CS Press, 1996.

[14] "MALLET: A Machine Learning for Language Toolkit." A. K. McCallum. http://mallet.cs.umass.edu. 2002.

[15] "Gestalt: Integrated Support for Implementation and Analysis in Machine Learning Processes." Patel, K., Bancroft, N., Drucker, S., Fogarty, J., Ko, A., Landay, J.A. Proceedings of the ACM Symposium on User Interface Software and Technology (UIST), 2010.

[16] "The Need for Open Source Software in Machine Learning" Sören Sonnenburg, Mikio L. Braun, Cheng Soon Ong, Samy Bengio, Leon Bottou, Geoffrey Holmes, Yann LeCun, Klaus-Robert Müller, Fernando Pereira, Carl Edward Rasmussen, Gunnar Rätsch, Bernhard Schölkopf, Alexander Smola, Pascal Vincent, Jason Weston, Robert Williamson; 8:2443--2466, 2007.

[17] "Workflows for e-Science: Scientific Workflows for Grids," Ian J. Taylor, Ewa Deelman, Dennis B. Gannon, Matthew Shields (Editors), Springer, January 2007

[18] "Weka: Practical Machine Learning Tools and Techniques with Java Implementations," I. H. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes, and S. J. Cunningham, ICONIP/ANZIIS/ANNES, pp. 192–196, 1999.

[19] "A Comparative Study on Feature Selection in Text Categorization." Yang, Y., Pedersen, J. O. International Conference on Machine Learning, 412–420, 1997.