DAFNA – RestFul API for Truth Discovery Algorithms

API DOCUMENTATION

M. Lamine Ba, Hossam. H. Hammady, Laure Berti-Equille

Qatar Computing Research institute
Hamad Bin Khalifa University

Creation date: November 5, 2015
Revision date: December 7, 2015

Contents

1 Introduction

Data Forensics with Analytics, or for short DAFNA, is a project initiated by the Data Analytics Group in Qatar Computing Research Institute, Hamad Bin Khalifa University. One of its various facets aims at providing efficient algorithms and tools for verifying the veracity of information provided by multiple conflicting sources. The DAFNA API for truth discovery is a RestFul API that enables end user applications to access and use twelve existing truth discovery algorithms for fact checking and truth discovery when they have to integrate heterogeneous information coming from untrustworthy sources. This documentation details the use of the API for third-party applications.

If you use this API or related achievements such as the online tool AllegatorTrack, please do not forget to aknowledge the authors by citing "Dalia Attia Waguih, Naman Goel, Hossam M. Hammady, Laure Berti-Equille. AllegatorTrack: Combining and Reporting Results of Truth Discovery from Multi-source Data. The 31th International Conference on Data Engineering (ICDE), Seoul, Korea, 2015".

2 Authentication

Third party developers wanting to test DAFNA API for truth discovery, or integrate it in their own applications, need to first get the necessary access credentials. To obtain credentials, every 3rd party developer must follows the process below.

- Go to AllegatorTrack
- Register using a valid email id and email confirmation
- Contact DAFNA developers for token access.

Upon token access request done, the user will receive from DAFNA developpers the following access credentials.

- **Authentication Token**: API user's access key
- **Encrypted Password**: user's secret key

3 Dataset Format

All the datasets used as input through the API must be CVS files having one of the two following header lines (the order between column names is important and must be respected).

1. **ObjectID, PropertyID, PropertyValue, SourceID, TimeStamp**

For instance, given a flight dataset, we have using (1):

"**ObjectID**", "**PropertyID**", "**PropertyValue**", "**SourceID**", "**TimeStamp**"
"AA-1623-EWR-MIA2011-12-18", "ActualDepartureTime", "12/18/11 2:58 PM (-05:00)", "myrateplan", "2011-12-18"
"AA-1623-EWR-MIA2011-12-18", "ActualArrivalTime", "12/18/11 5:14 PM (-05:00)", "myrateplan", "2011-12-18"
"AA-1623-EWR-MIA2011-12-18", "ActualDepartureTime", "12/18/11 2:58 PM (-05:00)", "helloflight", "2011-12-18"
"AA-1623-EWR-MIA2011-12-18", "ActualArrivalTime", "12/18/11 5:14 PM (-05:00)", "helloflight", "2011-12-18"

2. **ObjectID, PropertyID, PropertyValues, SourceID, TimeStamp**

Given a book dataset with a property having multiple values, we obtain using (2):

"**ObjectID**", "**PropertyID**", "**PropertyValues**", "**SourceID**", "**TimeStamp**"
"0023606924", "AuthorName", "schaefer. marcus, johnsonbaugh. richard", "a1books", "null"
"0023606924", "AuthorName", "schaefer. marcus, johnsonbaugh. richard", "alinonline", "null"
"0023606924", "AuthorName", "schaefer, marcus, johnsonbaugh. richard", "california textbooks", "null"
"0023606924", "AuthorName", "johnsonbaugh. richard", "deepak sachdeva", "null"

If the dataset adopts the second type of header line, the system will automatically split the contents of this column on the comma when running multi-valued algorithms.

As for the ground truth datasets they must be a CSV file with a header line having the following column names in any order: **ObjectID, PropertyID, PropertyValue**. Note that the values in **ObjectID** should match those in the corresponding dataset file so that objects are correctly matched (case sensitive). Again, you can replace **PropertyValue** by **PropertyValues** and the system will automatically split the contents of this column on the comma when running multi-valued algorithms. Below is an example of a ground truth dataset formatting.

"**ObjectID**", "**PropertyID**", "**PropertyValue**"
"AA-1221-MCO-ORD2011-12-01", "ExpectedDepartureTime", "12/01/2011 08:00 PM"
"AA-1221-MCO-ORD2011-12-01", "ActualDepartureTime", "12/01/2011 08:23 PM"
"AA-1221-MCO-ORD2011-12-01", "DepartureGate", "17"
"AA-1221-MCO-ORD2011-12-01", "ExpectedArrivalTime","12/01/2011 09:45 PM"

4 Request Format

4.1 Scheme

Requests to the API endpoints are done using plain HTTP. HTTPS is planned to be supported in the future development of the API for ensuring a more secure transmission of the Access Token.

4.2 Methods

The current version of DAFNA API supports three HTTP Requests, i.e., **GET**, **POST**, and **DELETE**, with respect to the operations, for instance creation, listing, or removal of datasets, the 3rd party applications would like to perform on the behalf on the client.

4.3 Base URL

The API is accessible through the following base URL.

- http://dafna.qrci.org/

Every endpoint of the API must be appended to this given base URL. For instance, to list the set of available datasets, client must use the following URL.

GET http://dafna.qcri.org/datasets/

4.4 Request Content Type

Either (1) “application/x-www-form-urlencoded” or (2) “application/json” can be used as the request content type (HTTP request “Content-Type” header value). If (1) is used, all parameters should be encoded in the URL as a query string with proper escapes and separated by &. If (2) is used, the request body should be a well-formed JSON object.

1. **url-encoded form**

   POST http://dafna.qcri.org/runsets?checked_algo=array_of_parameters

2. **json encoding**

   POST http://dafna.qcri.org/runsets {checked_algo: array_of_parameters, general_config: array_numerical_values}

Note that when using some command line tools to issue your requests, the way the body content is passed could be slighted different. For example, when a particular user issues requests through command lines via “Curl”, (1) and (2) argument passing techniques are respectively implemented as follows.

curl -X DELETE http://dafna.qcri.org/runsets/1/ -d “user_token=some_string_here”

and

curl -X POST http://dafna.qcri.org/runsets -H “Content-type: application/json" -d ‘{“user_token”: “some_string_here”, “datasets”: {“dataset_id”:“1”, “gt_id”:“1”}, “checked_algo”: {“Accu”: [“0.2”, “0”, “100”, “0.5”, “false”, “true”, “true”, “false”] }, “general_config”:[“0.001”, “0.8”, “1”, “0.4”]}’

5 Response Format

5.1 HTTP Status Codes

- **200:** Successful Requests (otherwise, an internal error will occur)
- **401:** Unauthorized Access, e.g., missing, invalid, or expired token
- **403:** Forbidden Access to the resource requested by the client
- **404:** Not Found Resource requested by the user
- **422:** Invalid Action on the accessed resource

5.2 Response Body Format

Responses to user queries are returned in JSON format. For now, JSON is the only supported format by the API for query output. Further development might consider other formats such as XML.

6 Endpoints

The current version of the API provides 8 endpoints, 4 among them are accessed via GET, 2 via POST, and 2 via DELETE:

6.1 POST Endpoints

1. **/datasets**

   Requests the creation of a Dataset

   Parameters

   - **user_token**: secret token provided to the client for authentication
   - **s3_key**: a file's s3 key if the file is uploaded at Amazon's s3 official. Right access to this bucket is needed.
   - **other_url**: URL (other than the s3 key) if the dataset file is located somewhere else, e.g. in a local computer.
   - **kind**: either "claims" or "ground" corresponding to the different types of handled datasets
   - **original_filename**: name of the uploaded file. It corresponds to any name that you want to give for identification in future

   Response

   No data.

200 status code when the request succeeds

Notes

a) One of **s3_key** or **other_url** parameters are mandatory for parsing dataset.

b) All uploaded files should be encoded in **UTF-8**.

c) Be sure that your dataset fulfills the required format as described in Section 3.

2. **/runsets**

Creating a Runset, i.e., a truth discovery job over a given dataset

Parameters

- **user_token**: secret token provided to the client for authentication
- **datasets**: list of dataset identiers referring to the datasets on which the runset should applied. An example input is **datasets**:{“203”:“1”, “204”:“1”} denoting the choice of the dataset numbered 203 along with the corresponding ground truth dataset with the number 204.
- **checked_algo**: set of algorithm name(s), each followed by an array of numerical values for its different input parameters, e.g., “checked_algo”: {“Accu”: [“0.2”, “0”, “100”, “0.5”, “false”, “true”, “true”, “false”] , “Cosine”: [“1”, “0.2”] } specifies two algorithms **Accu** and **Cosine**
- **general_config**: an array of numerical values, e.g., [“0.001”, “0.8”, “1”, “0.4”]

Response

No data.

200 status code when request succeed.

Notes

a) **checked_algo** allows to specify the set of truth discovery algorithms along with the values of their input arguments to be used for the truth discovery process. At least one algorithm must be provided, and when several algorithms are specified their results are combined by the system for a better accuracy. We detail in Section 7 the set of supported truth discovery algorithms together with their input arguments.

b) **general_config** allows to specify the values of the general parameters of a runset, i.e., the convergence test thresold and the initial sources’ trusworthiness scores.

6.2 GET Endpoints

1. **/datasets**

Listing Datasets

Parameters

- **user_token**: secret token provided to the client for authentication
- **kind**: either “claims” or “ground” which correspond to the different types of available datasets
- **start**: an integer number specifying the index from which the listing of available datasets must start
- **length**: an integer specifying the maximum number of datasets to be listed

Response

{DS_1, DS_2, . . ., DS_n} where every DS_n is formatted as follows.

{ draw: <draw>, recordsTotal: <Number of Records>, recordsFiltered: <Number of filtered Records>, data: <data>, s3_direct_post: <date_time>, fields: <date_time>, key: <date_time>, policy: <policy>, signature: <signatur>, success_action_status: <status> }

where <data> contains all the information about to the available datasets.

Notes

a) **start** and **length** parameters are optional.

b) When the parameter **kind** is not provided, the user will get an empty set of datasets.

c) When the arguments **start** and **length** are not specified, all the available datasets of the specified kind are returned.

2. **/runsets**

Listing Runsets

Parameters

- **user_token**: secret token provided to the client for authentication

Response:

{RS_1, RS_2, $\cdots$, RS_n} where the format of RS_n is:
{ id: <runset id>, created_at: <Time of creation>, runs:[{ id: <runset id>, algo: <Time of creation>, crated_at: <>, status: <whether finished>, duration: <Duration of run>, } ,{},{}...] }

3. **/runsets/<id>/results**

Listing Results of runsets having id=<id>

Parameters

- **user_token**: secret token provided to the client for authentication
- **id**: an existing identier of the runset whose results should be listed

Response

{ draw: <draw>, recordsTotal: <Total Number of Records>, recordsFiltered: <Filtered Records>, data:

}

4. **/runs**

   Listing Runs

   Parameters

   - **user_token**: secret token provided to the client for authentication

   Response

   { draw: <draw>, recordsTotal: <Total Number of Records>, recordsFiltered: <Filtered Records>, data:[{ id: <run id>, algorithm: <Name of the algorithm>, created_at: <Time of creation>, runset_id: <Runset ID>, display: <Algorithm name and parameters in paranthesis (in order)>, status: <whether finished>, duration: <Duration of run>, } ,{},{}...] }

6.3 DELETE Endpoints

1. **/datasets/<id>**

   Deleting a Dataset having id=<id>

   Parameters

   - **user_token**: secret token provided to the client for authentication
   - **id**: an existing identier of the dataset to be deleted

   Response

   No data.

   200 status code when the request succeeds.

2. **/runsets/<id>**

   Deleting a Runset having id=<id>

   Parameters

   - **user_token**: secret token provided to the client for authentication
   - **id**: an existing identier of the runset to be deleted

   Response

   No data.

   200 status code when the request succeeds.

7 Supported Algorithms

The main interest of using the API is to create a truth discovery job which is possible with an instantiation of a runset. Such a instantiation requires to specify one or more truth discovery algorithms to use through the argument **checked_algo**. In the following, we detail the twelve truth discovery algorithms and their specific input parameters supported by DAFNA API.

7.1 Cosine

Description: Cosine is a heuristic approach for estimating a value confidence and source trustworthiness, based on the cosine similarity measure proposed by A. Galland, S. Abiteboul, A. Marian, and P. Senellart. Corroborating Information from Disagreeing Views. In WSDM, pp. 131–140, 2010.

Parameters

- **Initial Value Confidence**: Initialization of value confidence for all properties values.

  dataType: double, **min-value:** 0, **max-value:** 1.0, **default-value:** 1

- **Prediction constant**: Constant that gives more weight to predictable views (eg, sources with consistently often correct claims or consistently often wrong claims).

  dataType: double, **min-value:** 0, **max-value:** 1, **default-value:** 0.2

7.2 2-Estimates

Description: 2-Estimates is a probabilistic model built over the heuristic model, Cosine and proposed by A. Galland, S. Abiteboul, A. Marian, and P. Senellart. Corroborating Information from Disagreeing Views. In WSDM, pp. 131–140, 2010.

Parameters

- **Normalization Factor**: The weight used for normalizing sources' trustworthiness and values' confidence.

  dataType: double, **min-value:** 0, **max-value:** 1, **default-value:** 0.5

7.3 3-Estimates

Description: 3-Estimates extends 2-Estimates proposed by A. Galland, S. Abiteboul, A. Marian, and P. Senellart. Corroborating Information from Disagreeing Views. In WSDM, pp. 131–140, 2010.

Parameters

- **Initial Error Factor**: Initialization of value's error factor for all properties values.

  dataType: double, **min-value:** 0, **max-value:** 1.0, **default-value:** 0.4

- **Normalization Factor**: The constant used for normalizing source truthworthiness and value confidence.

  dataType: double, **min-value:** 0, **max-value:** 1, **default-value:** 0.5

7.4 Depen

Description: First Bayesian truth discovery model that takes into cosideration the copying relationship between sources, proposed by X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: The role of source dependence. PVLDB, 2(1):550–561, 2009.

Parameters

- **Dependence priori probability**: The a priori probability that two data sources are dependent.

  dataType: double, **min-value:** 0, **max-value:** 0.5, **default-value:** 0.2

- **Copied value probability**: The probability that a value provided by a copier is copied.

  dataType: double, **min-value:** 0, **max-value:** 1, **default-value:** 0

- **Number of false values**: The number of false values in the underlying domain for each object.

  dataType: int, **min-value:** 1, **max-value:** 1, **default-value:** 100

- **Similarity Constant**: The similarity betwen values is weighted for the value confidence computation.

  dataType: double, **min-value:** 0, **max-value:** 1, **default-value:** 0.5, **hidden:** True

- **considerSimilarity**: False for Depen: the model does not take into account value similarity.

  dataType: boolean, **min-value:** 0, **max-value:** 1, **default-value:** False, **hidden:** True

- **considerSourcesAccuracy**: False for Depen: the model does not take into account the accuracy of the sources.

  dataType: boolean, **min-value:** 0, **max-value:** 1, **default-value:** False, **hidden:** True

- **considerDependence**: True for Depen: the model takes into account the dependence between sources.

  dataType: boolean, **min-value:** 0, **max-value:** 1, **default-value:** True, **hidden:** True

- **orderSrcByDependence**: If true, Depen model is based on source dependence ordering, otherwise on lexicographic ordering.

  dataType: boolean, **min-value:** 0, **max-value:** 1, **default-value:** False, **hidden:** True

7.5 Accu

Description: Accu extends Depen model and considers the source accuracy in addition to the source dependence, proposed by X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: The role of source dependence. PVLDB, 2(1):550–561, 2009.

Parameters

- **Dependence a priori probability**: The a priori probability that two data sources are dependent.

  dataType: double, **min-value:** 0, **max-value:** 0.5, **default-value:** 0.2

- **Copied value probability**: The probability that a value provided by a copier is copied.

  dataType: double, **min-value:** 0, **max-value:** 1, **default-value:** 0

- **Number of false values**: The number of false values in the underlying domain for each object.

  dataType: int, **min-value:** 1, **max-value:** 1, **default-value:** 100

- **Similarity Constant**: The similarity betwen values is weighted for the value confidence computation.

  dataType: double, **min-value:** 0, **max-value:** 1, **default-value:** 0.5, **hidden:** True

- **considerSimilarity**: False for Accu: the model does not take into account value similarity.

  dataType: boolean, **min-value:** 0, **max-value:** 1, **default-value:** False, **hidden:** True

- **considerSourcesAccuracy**: True for Accu: the model takes into account the accuracy of the sources.

  dataType: boolean, **min-value:** 0, **max-value:** 1, **default-value:** True, **hidden:** True

- **considerDependency**: True for Accu: the model takes into account the dependence between sources.

  dataType: boolean, **min-value:** 0, **max-value:** 1, **default-value:** True, **hidden:** True

- **orderSrcByDependence**: If true, Accu model is based on source dependence ordering, otherwise on lexicographic ordering.

  dataType: boolean, **min-value:** 0, **max-value:** 1, **default-value:** False, **hidden:** True

7.6 AccuSim

Description: Accusim extends Accu model and considers the value similarity, proposed by X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: The role of source dependence. PVLDB, 2(1):550–561, 2009.

Parameters

- **Dependence a priori probability**: The a priori probability that two data sources are dependent.

  dataType: double, **min-value:** 0, **max-value:** 0.5, **default-value:** 0.2

- **Copied value probability**: The probability that a value provided by a copier is copied.

  dataType: double, **min-value:** 0, **max-value:** 1, **default-value:** 0

- **Number of false values**: The number of false values in the underlying domain for each object.

  dataType: int, **min-value:** 1, **max-value:** 1, **default-value:** 100

- **Similarity Constant**: The similarity betwen values is weighted for the value confidence computation.

  dataType: double, **min-value:** 0, **max-value:** 1, **default-value:** 0.5

- **considerSimilarity**: True for AccuSim: the model takes into account value similarity.

  dataType: boolean, **min-value:** 0, **max-value:** 1, **default-value:** True, **hidden:** True

- **considerSourcesAccuracy**: True for AccuSim: the model takes into account the accuracy of the sources.

  dataType: boolean, **min-value:** 0, **max-value:** 1, **default-value:** True, **hidden:** True

- **considerDependency**: True for Accusim: the model takes into account the dependence between sources.

dataType: boolean, **min-value:** 0, **max-value:** 1, **default-value:** True, **hidden:** True

- **orderSrcByDependence**: If true, AccuSim model is based on source dependence ordering, otherwise on lexicographic ordering.

  dataType: boolean, **min-value:** 0, **max-value:** 1, **default-value:** False, **hidden:** True

7.7 AccuNoDep

Description: AccuNoDep extends Accu model while assuming all sources are independent, proposed by X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: The role of source dependence. PVLDB, 2(1):550–561, 2009.

Parameters

- **Dependence a priori probability**: The a priori probability that two data sources are dependent.

  dataType: double, **min-value:** 0, **max-value:** 0.5, **default-value:** 0.2

- **Copied value probability**: The probability that a value provided by a copier is copied.

  dataType: double, **min-value:** 0, **max-value:** 1, **default-value:** 0

- **Number of false values**: The number of false values in the underlying domain for each object.

  dataType: int, **min-value:** 1, **max-value:** 1, **default-value:** 100

- **Similarity Constant**: The similarity betwen values is weighted for the value confidence computation.

  dataType: double, **min-value:** 0, **max-value:** 1, **default-value:** 0.5, **hidden:** True

- **considerSimilarity**: False for AccuNoDep: the model does not take into account value similarity.

  dataType: boolean, **min-value:** 0, **max-value:** 1, **default-value:** False, **hidden:** True

- **considerSourcesAccuracy**: True for AccuNoDep: the model takes into account the accuracy of the sources.

  dataType: boolean, **min-value:** 0, **max-value:** 1, **default-value:** True, **hidden:** True

- **considerDependency**: False for AccuNoDep: the model considers that all sources are independent.

dataType: boolean, **min-value:** 0, **max-value:** 1, **default-value:** False, **hidden:** True

- **orderSrcByDependence**: If true, AccuNoDep model is based on source dependence ordering, otherwise on lexicographic ordering.

  dataType: boolean, **min-value:** 0, **max-value:** 1, **default-value:** False, **hidden:** True

7.8 TruthFinder

Description: TruthFinder, proposed by X. Yin, J. Han, and P. S. Yu. Truth Discovery with Multiple Conflicting Information Providers on the Web. TKDE, 20(6):796–808, 2008, is a Bayesian model that iteratively computes and updates the trustworthiness of a source as a function of the belief in its claims, and then the belief score of each claim as a function of the trustworthiness of the sources asserting it.

Parameters

- **Similarity Constant**: The similarity betwen values is weighted for the value confidence computation.

  dataType: double, **min-value:** 0, **max-value:** 1, **default-value:** 0.5

- **Dampening Factor**: The damping factor compensates the effect when sources with similar values are actually dependent.

  dataType: double, **min-value:** 0, **max-value:** 1, **default-value:** 0.1

7.9 SimpleLCA

Description: SimpleLCA (Latent Credibility Analysis) is a probabilistic model proposed by J. Pasternack and D. Roth. Latent Credibility Analysis. In WWW, pp. 1009–1020, 2013. Each source has a probability of being honest, and all sources are considered independent.

Parameters

- **Prior truth probability**: Prior probability of a claim being the true claim.

  dataType: double, **min-value:** 0, **max-value:** 1, **default-value:** 0.5

7.10 GuessLCA

Description: GuessLCA extends SimpleLCA by adding a probability of a source guessing the true value while being honest.

Parameters

- **Prior truth probability**: Prior probability of a claim being the true claim.

  dataType: double, **min-value:** 0, **max-value:** 1, **default-value:** 0.5

7.11 MLE

Description: MLE (Maximum Likelihood Estimation) proposed by D. Wang, L. M. Kaplan, H. K. Le, and T. F. Abdelzaher. On Truth Discovery in Social Sensing: a Maximum Likelihood Estimation Approach. In IPSN, pp. 233–244, 2012, is based on the Expectation Maximization (EM) algorithm to quantify the reliability of sources and the correctness of their observations. It only deals with boolean positive attributes observations.

Parameters

- **Prior truth probability**: Overall prior probability of the claims being the true claims.

  dataType: double, **min-value:** 0, **max-value:** 1, **default-value:** 0.5

- **r**: The probability that a source provides a value for all data items.

  dataType: double, **min-value:** 0, **max-value:** 1, **default-value:** 0.5

7.12 LTM

Description: LTM (Latent Truth Model) proposed by B. Zhao, B. I. P. Rubinstein, J. Gemmell, and J. Han. A Bayesian Approach to Discovering Truth from Conflicting Sources for Data Integration. PVLDB, 5(6):550–561, 2012, uses Bayesian Networks for estimating the truth. LTM iterates over a fixed number of iterations, and compute the value confidence using Collapsed Gibbs Sampling process.

Parameters

- **Prior truth probability**: Prior probability of how likely each claim is to be true.

  dataType: double, **min-value:** 0.0, **max-value:** 1.0, **default-value:** 0.5

- **Prior falsehood probability**: Prior probability of how likely each claim is to be false.

  dataType: double, **min-value:** 0.0, **max-value:** 1.0, **default-value:** 0.5

- **Prior true negative count**: Prior number of true negatives for the sources.

dataType: double, **min-value:** 0.0, **max-value:** 1.0, **default-value:** 0.1

- **Prior false positive count**: Prior number of false positives for the sources.

  dataType: double, **min-value:** 0.0, **max-value:** 1.0, **default-value:** 0.9

- **Prior false negative count**: Prior number of false negatives for the sources.

  dataType: double, **min-value:** 0.0, **max-value:** 1.0, **default-value:** 0.1

- **Prior true positive count**: Prior number of true positives for the sources.

  dataType: double, **min-value:** 0.0, **max-value:** 1.0, **default-value:** 0.9

- **Number Of Iterations**: Number of Iterations.

  dataType: int, **min-value:** 1, **max-value:** 10000, **default-value:** 500

- **Burn-in**: Collapsed Gibbs Sampling burn-in period (ie, the number of discarded first set of iterations) must be significantly less than the number of iterations.

  dataType: int, **min-value:** 0, **max-value:** 1000, **default-value:** 100

- **Thinning**: Collapsed Gibbs Sampling thinning parameter (ie, the number of iterations to be skipped every time before considering the result of a selected iteration) must be significantly less than the number of iterations.

  dataType: int, **min-value:** 0, **max-value:** 1000, **default-value:** 9

†**General Parameters.** Parameters needed for all Truth Discovery Algorithms (specified via the **general_config** parameter)

Parameters

- **Convergence test threshold**: The difference of source truthworthiness cosine similarity between two successive iterations should be less than the user-defined threshold.

  dataType: double, **min-value:** 0, **max-value:** 0.1, **default-value:** 0.001, **step:** 0.001

- **Initial sources truthworthiness**: Initialization of all sources truthworthiness.

  dataType: double, **min-value:** 0, **max-value:** 1.0, **default-value:** 0.8

- **Initial Value Confidence**: Initialization of value confidence for all properties values.

  dataType: double, **min-value:** 0, **max-value:** 1.0, **hidden:** True

- **Initial Error Factor**: Initialization of error factor for all properties values.

  dataType: double, **min-value:** 0, **max-value:** 1.0, **default-value:** 0.4, **hidden:** True

†**Combiner.** The combiner of selected truth discovery algorithms. It is automatically selected if you select more than 1 algorithm and 1 ground truth dataset. It is based on Bayes combination from P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, 29:103–130, 1997.

8 DAFNA API Usage: Use Case

Let us consider a typical scenario where a given user, holding an access token **"_XYH128PL04"** to the DAFNA API, would like to discover the correct list of authors of a set of books. The user has first extracted the candidate authors of these books from different Web sources and has stored them into a csv file **book.csv** in a local directory named **./datasets/**. To discover the truth using our API, the user needs to upload (or create) the dataset, get the dataset identifier, create a runset (using **Accu** algorithm), get the runset identifier, and finally view the result of this runset. The user can further decide to delete the copy of the dataset on the server. To this end, she (or he) should issues following sequence of requests.

1. curl -X POST http://dafna.qcri.org/datasets -H "Content-Type: application/x-www-form-urlencoded; charset=UTF-8" -d "user_token=_XYH128PL04" -d "other_url=./dataset/book.csv" -d "original_filename=book.csv&kind=claims" (upload the dataset of the server)

2. curl -X GET http://dafna.qcri.org/datasets -d "user_token=_XYH128PL04" -d "kind=claims" -d "start=0" -d "length=54" (list the set of datasets on the server: let's assume that this returns the book dataset with the identifier **"120"**)

3. curl -X POST http://dafna.qcri.org/runsets -H "Content-type: application/json" -d '{"user_token": "_XYH128PL04", "datasets": {"120":"1"}, "checked_algo": {"Accu": ["0.2", "0", "100", "0.5", "false", "true", "true", "false"] }, "general_config":["0.001", "0.8", "1", "0.4"]}' (execute a truth discovery on book.csv using **Accu** algorithm)

4. curl -X GET http://dafna.qcri.org/runsets -d “user_token=_XYH128PL04” (list the set of current runsets: let’s assume it returns our runset with the identier **“1”**)

5. curl -X GET http://dafna.qcri.org/runsets/1/results -d “user_token=_XYH128PL04” (view the result of the truth discovery process over book.csv)

6. curl -X DELETE http://dafna.qcri.org/datasets/120 -d “user_token=_XYH128PL04” (remove the copy of the dataset on the server)

† Do not forget to provide a valid token access and a properly formatted dataset if you want to the given use case.

9 Job Monitoring using Pusher

Some of the jobs you submit (e.g. uploading datasets or running algorithm(s)) might take time to complete for various reasons. Instead of letting users periodically pinging the server to verify the status of their ongoing jobs, we provide a manner to make things easier for user applications by setting up a channel on Pusher. User applications can subscribe to this channel in oder to be automatically notify for the events they have been binded to. When any event like task completion occurs, a user applicaton will be notified of this. Kindly note that the unexpected approach of repeatedly requesting run statuses rather than listening to the push messages will eventually overwhelm our server, and we may choose to suspend your account. So, please go through the documentation of Pusher to get a sense of how things work. An client can subscribe to the notifications for her application using **app_id = “79143”**. To this end, a client needs to contact us for a key, which will be sent by email, to enable him to subscribe to the channel. Once you have configured your application using the app_id and key, you can bind to particular events (see examples below), you wish to be notified for.

- **channel.bind(‘run_change’, function(run)):** will notifiy in case of any change in the status of a run.
- **channel.bind(‘dataset_change’, function(dataset)):** will notify you in case of any change in the status of a dataset.

You may find the following pages to be particularly useful:

1. Client Channel
2. Client Public Channels
3. Client Connect
4. Client Events

10 Online Tool: AllegatorTrack

The main goal of DAFNA project is to design a scalable and accurate truth discovery system to score the veracity of information extracted from multiple online sources. We design DAFNA API as the first proposed API that allows users to tests and compares a large class of well-known truth finding algorithms on different scenarios and datasets. AllegatorTrack, whose the interface is depicted in Figure 1, is the first application that uses DAFNA API and provides to users an online plateform for testing the different features of the API. AllegatorTrack application is accessible at: http://dafna.qcri.org/users/sign_in.

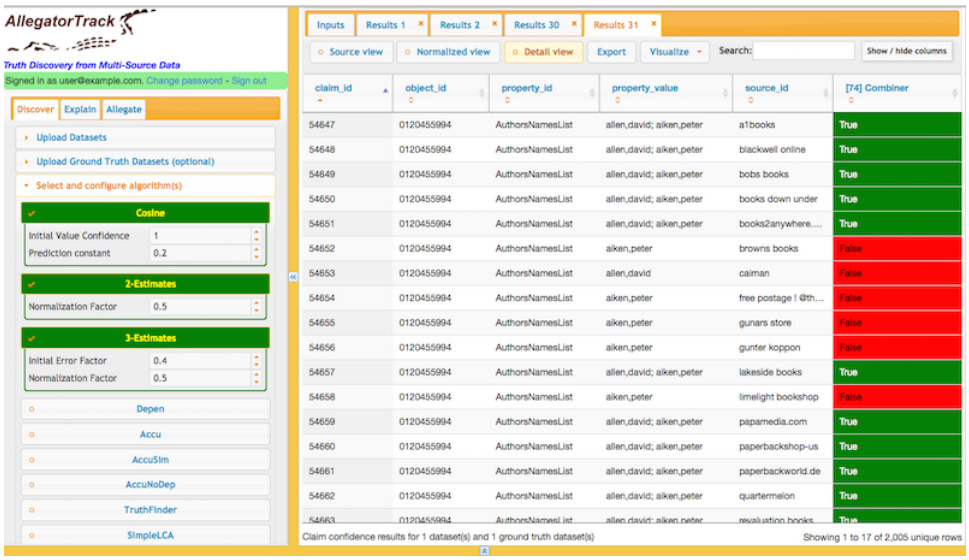

Figure 1: AllegatorTrack System

11 Further Developments

Currently, the API is still in improvements and many new other aspects of truth finding and endpoints are still under investigation, based on recent proposed development in the field, discussions with early adopter developers, and users' feedbacks. For example, the scheme may support HTTPS in the future, the response body format may support other output formatting like XML.Most importantly, the API must be extended in the future in order to propose broader class of truth finding algorithms, in particular the new ones.

12 Questions or Feebacks

For any questions or feebacks about the API, please feel free to contact us or visit the website of the DAFNA project for more details by following the link below: http://dafna.qcri.org.