OEID for Sentiment Analysis: tell me your #hashtag!

Maria Emanuela Arico
Ettlingen

Schlüsselworte
Endeca, OEID, ICS, Discovery Information, Sentiment Analysis, Twitter

Data Discovery & Social Network

Nowadays every day we generate tons of data, from different sources, also doing daily and, for us, simple things. Just let's think, for example, about the data generated by our mobile phones, all the information related to the network, about our position, all the data sent and received to follow our social network and so on. Data Discovery is a term used to generalize all the technics that are used to collect data from multiple external sources, organize them and explore them to discover something new. In our case, we are going to apply these technics in order to collect data from social network, analyse them and discover new information and opinions.
This one is not the only way to apply information discovery to real life. We could do a lot of examples, starting from fraud analysis taking data from different sources, competitor analysis, using also e-commerce website, or for example customer targeting, in order to analyse our customer and offer them specific services.

Oracle Endeca Information Discovery

The Oracle solution for Data Discovery is Oracle Endeca Information Discovery, a suite composed by different pieces:

- Endeca Server
- Endeca Studio
- Endeca Integrator
- Web Acquisition Tool (WAT)

Endeca Information Discovery is able to analyse data coming from structured and not-structured data sources, give us the possibility to discover new information and take new decisions. It is important to notice that Endeca is not a replacement for the Oracle Business Intelligence Suite, but another tool that allows us to get another type of analysis. With Oracle Business Intelligence we are able to get answers to questions we already know (so basically we know what we are searching for). With Endeca we are able instead to understand which ones are our new questions and get answers to them.

The core of the Endeca Suite is the Endeca Server. It is based on the MDEX Engine, that is able to store big amount of data by using a vertical record store that saves them in a columnar database using two different index: the value and a unique key. To improve the speed of the research the information is stored in cache using a tree-structure for the index, and it saves also the most requested query.

Endeca Studio is a front-end application, available from any device via a web browser, that allow us to create applications by using several pre-built components in order to explore our data. We have also the possibility to create some fast stand-alone applications, using for example some excel files as source.

Oracle Endeca Integrator is the development environment used to load, enrich and organize the information coming from our data-source to load them in the Endeca Server. With some plugins we are able to add also the Sentiment Analysis, add tags to our data, create and manage multi-value field, called MultiAssign (like if one value was an array with the advantage that the values inside are managed as single ones).

Web Acquisition Tool (WAT) is a development environment that combines the user friendly aspects of a web browser with a visual flow editor to take information from different kind of data sources, as web sites, without the need to write code.

Now that we know the elementary structure of Oracle Endeca we can talk deeply about the features we are going to use.

Sentiment Analysis

In 2015 we spent an average of 106,2 minutes every day on social network. And every second an average of 6000 tweets are published on Twitter.
This little information can make us understand the power of the social networks nowadays.
We use it to say how we feel, to share information and also to give feedback about our shopping.
Sentiment Analysis is basically the successor of the classic opinion survey.
Sentiment Analysis allows us to catch and understand if users' opinions are positive or not.
In our case we are going to use Twitter as data source. Thanks to Twitter development API, doing this kind of experiments is very easy. We will use the Twitter Search API to send a request and get back tweets that matches our research to process them in our Sentiment Engine. The Oracle Endeca plugin that allows us to do this is called Oracle Endeca Text Enrichment with Sentiment Analysis, and we will use it inside Oracle Endeca Integrator with other native components to extract, elaborate and load inside the Oracle Endeca Server our data.
Endeca Text Enrichment is based on the comparison between an internal dictionary and the source. Each entry of the dictionary has a score that can be positive (greater than 0), neutral (equal to 0), or negative (less than 0).

Let's Analyse Hashtags!

We are going to analyse the hashtag of this event, which is #doag2016.
In order to have our data source we are going to use the Twitter Search API to get several .json files with all the tweets matching our query.
A tweet grabbed by the Twitter search API has this structure:

Requesting a .json file from command line is quite simple: we need just to have a developer account in Twitter and request the OAuth Signature from the OAuth Tool. In our case, we are going to use a GET request type, using the URI of the search API (https://api.twitter.com/1.1/search/tweets.json). The query must have this structure:
q=%23doag2016&result_type=recent&lang=de

We are researching for the hashtag #doag2016, among all the recent tweets in English. If we want to have a preview of our research we can easily simulate it in our internet browser, pasting our request query after the address *https://twitter.com/search?* .
In our case, our "test address" would be
https://twitter.com/search?q=%23doag2016&result_type=recent&lang=de

We will get back a cURL to copy-paste in our terminal, and we are going to save the result of this request in a .json file. At the end of our .json file we will find a last string that will tell us if we have more tweets to grab or not. We will find this information after the “next_result” tag, with the new query to use. When there are no more tweets, we will find only the information about the completed operation under the tag “completed_in”. After this we will move our .json file in the folder “data-in” of our Endeca Project.

Before starting with our graph, we have to set and fix some configuration files.
First of all we will configure the workspace.prm, with some project variable we will use the components of our graphs, information about the connection to the Endeca Server, the name of data-domain, directory of our Endeca Text Enrichment and other informations. The next configuration file we will edit will be the salience.properties with some information such as the Entity and Theme extraction and the correct userdata directory to be used.

First of all, we will run the InitDataDomain graph to set properly our data-domain. The graph will check the status of the data-domain, check the name we inserted inside our workspace.prm: if it does not exist it will create a new one with the name we chose, if it exists but is disabled, Endeca Integrator will enable it and if the data-domain exists and is enabled, Endeca Integrator will do nothing.
After this, we will execute the LoadConfiguration graph that will set and load some configuration of our data-domain. The configuration files that are used for this step are stored in the folder “config-in” of our project.
The main files for the configurations are:

- The AttributeFormatting.csv, stores the information about some default characteristics for some type of format. The allowed syntax is the following:

Format, AvailableAggregations,DefaultAggregations,Description,FormatSettings.

- The AttributeGroups.csv file, stores the information about the fields and in which groups we are going to organize them. The structure of this file is the following:

GroupName, GroupDisplayName, AttributeName

- The AttributeProfiles.csv, stores the information about the profiles we are going to use for our fields. The sintax is the following:

Profile,SelectType, RankingType,ShowRecordCounts,IsValueSearchable,IsRecordSearchable,SnippetSize,IsUnique,IsSingleAssign,IsDimension.
This setting is quite important in order to regulate the performance of our application: in phase of analysis (especially for big projects) it is fundamental to estimate in advance which are the fields we need as dimensions. This fields have to be indexed in our Endeca Server (Value/Record searchable), or we risk to have a slow application since we are also indexing and searching, in the Endeca Studio, for fields we do not need (such as an id or a phone number). In this moment we can also decide if our field is MultiAssign or SingleAssign.

- The AttributeMetadata.csv file, stores the detailed information about the metadata of our fields. The format of this file is the following:

AttributeKey, DisplayName, Type, Profile, Format.
A particular attention has to be given to the Type, Profile and Format settings.

The Type will decide in which kind of mdex type we will store that field. We can set:

- **string**, that corresponds to the Oracle type VARCHAR2; boolean, that corresponds to the Oracle type NUMBER (but Endeca accepts only value sets to "true/false" or "0/1";
- **dateTime**, that corresponds to the Oracle type TIMESTAMP (the accepted pattern is "yyyy '-' mm '-' dd 'T' hh ':' mm ':' ss {'.' s+} Z");
- **double**, that corresponds to the Oracle type NUMBER;
- **duration**, that corresponds to the Oracle type VARCHAR2 (since there is not specific type for the duration in Oracle, we have to give a string. The accepted pattern is "P' {d 'D'} 'T' {h 'H'} {m 'M'} {s {'.' s+} 'S'}");
- **geocode**, that corresponds to the Oracle type VARCHAR2 (as for the duration, there is no specific type for the geocode in Oracle, and we will have to give a string. The accepted format for the geocode is "latitude-value longitude-value";
- **int**, that corresponds to the Oracle type NUMBER;
- **long**, that corresponds also to the Oracle type NUMBER;
- **time**, that corresponds to the Oracle type VARCHAR2 (also for the time, there is no specific type in Oracle, and we will have to give a string. The accepted pattern is "hh ':' mm ':' ss {'.' s+} Z".

The Profile is linked to the profiles defined in the AttributeProfiles.csv.
The Format is linked to the formats defined in the AttributeFormatting.csv.

After sorting all our configuration files, we can run the graph LoadConfiguration, that is going just to load our configuration in the Data Domain in order to have it ready to get our source data in the next step. In this phase we are going to load also more information set in the workspace.prm.

Finally, we are ready for the big step, the one in which we will grab, elaborate, sort, enrich and load in the Endeca Server our Tweets.

We will analyse the main parts of our LoadData.grf.

First of all, we are going to use a component called JSONReader. We will not define a single file, but a complete folder, in order to let it read everything that will be inserted in that folder. This will be useful later. As folder we will give the Tweets folder we created inside data-in, and in which we pasted all the .json file extracted. In this step we have also to define the metadata of our source. This can be done automatically by Integrator (extracting directly from the structure of the source), but in our case we will do it manually.

After this we are going to sort our tweets for Tweet ID using the component FastSort, and to avoid duplicate tweets we will use the Dedup that will remove all the duplicate entries (in our case we will base this filter for the dedup on the Tweet ID). After this we are going to read another file, in which we are going to add some "important" profile on Twitter for our research. What we are going to do is using a Whitelist Tagging (basically we will extract some terms from the field we defined in order to create a multi-value attribute). The sintax of the file we will read has to be:
SearhTerm,TagValue
To apply this enrichment, we will use a component called Text Tagger Whitelist, putting as source first our tweets already sorted, and as second the file with our list of terms to tag.
Before starting our enrichment, we will apply some transformation in order to remove special characters and format some fields, and after we are going to apply the Text Enrichment for Sentiment Analysis. We will use a component called Text Enrichment, give as configuration file our

salience.propierties, decide which one is the field that has to be enriched (in our case the text of the tweet), and we can set some small more configurations. After this step, our data source is already enriched with the sentiment analysis data. After this, using a similar logic we used for the Whitelist Tagger before we are going to tag all the hashtag of the text. Before loading the source in our Endeca Server we are going to modify a bit the data to have them more formatted (putting fields as hashtag in lowercase, format the timedate of the tweet in a format accepted by the Endeca Server, etc) and we will load our data using a component called Endecs Bulk Add or Replace Records, that will take as configuration all the parameter we set in our configuration file (workspace.prm), the Collection Key (that the key of the collection in which we load our data) and the Spec Attribute (a sort of primary key).
Now we can execute our graph, and finally, we will have our data inside our data_domain, and we can now create/edit our application inside the Endeca Studio.
A small reminder has to be given about how Endeca manages the data in case we execute more times the LoadData.grf graph. As we can imagine, Endeca will load as "add" every time we execute the graph the data source we load. This means that in case we want just to reload the same data, we will have to execute a reset of the data_domain. To do this I recommend to have always prepared a "ResetDataDomain" graph.

Endeca Studio Application

After having our Data Domain populated, we can build a Studio Application. As we said at beginning, Studio is a visual editor, that using some prebuilt components, let us build easily and fast a working application (and can work also in a "stand alone" mode using for example an excel file). The most important feature is that all the components are updated in real time with the domain of our analysis. For our application I used some basic components that will be described now:

- The **Summarization Bar** is a metric container. We can use the default one given by Endeca, like count, sum, average, minimum or maximum for numeric fields. In the first metric in our application we can see the number of tweets in our domain of analysis. The second one is the total score for the Sentiment Analysis we got for the tweet we are analysing, and the third one is, instead, the average score of our Sentiment Analysis. We added also a small "conditional formatting" in order to give more visibility to the score: if the Sentiment Analysis is positive (greater than 0) it will be green, if it's neutral (equal to 0) it will be yellow and if it will be negative (less than 0) it will be red. This will give us an immediate idea about the sentiment for our domain of analysis.

- The **Tag Cloud** is basically a container for "keywords", showed with bigger or smaller font size depending on one chosen metric. In our case we will show the users that tweeted the most (so our metric will be the count of the tweets). This will allow us to identify immediately the most active users (in terms of number of tweets). We will do the same in the second Tag Cloud for the most tweeted Themes.

- We have different kinds of **Charts**, and we configure them as we want. In our case we are going to use different charts, to show us information about the tweet, the users, their location, etc.

- The **Result Table** is a table that shows us all the information inside our record. We can define several customized "views" and decide which ones are the fields we want to see. From this component we can also do an export our data.

- The **Result List** provides us a list of information in a similar way to a web search page. The field shown can be customized.
- The **Search Component** allow us to search words inside our data domain. The research is done only across the field we decide to set as searchable (*IsValueSearchable* in AttributeProfiles.csv). In the settings we are able to enable also the Boolean Research.
- From the **Available Refinements** we can see all the fields that we can use to refine our domain of analysis. We will see all the value in our data domain for that field, with the number of occurrences for it.
- All the filters applied in our domain of analysis are shown in the **Selected Refinements** component.

As we said before, all the components of one Studio Application follow the actual refinement: this means a real-time update of our components meanwhile we change the domain of our analysis, for example adding filters.

But… Not real time!

The application as we see, it's working fine. We have only a problem: it is not in real-time.
To have our data updated we will have to execute again manually our LoadData.grf.
What we will try, to avoid this problem, is integrating it with Oracle Integration Cloud Service: we will use it to integrate our Endeca Process.

ICS

Oracle Integration Cloud Service gives us the possibility to integrate our applications in the cloud, with intuitive graphic interfaces and no coding task.
With Oracle Cloud Integration Cloud we can manage our integrations without installing any software on our machines and it is fully compatible with Oracle SaaS applications.
Through the visual interface, the tool allows, with a drag and drop system, to add, manage and remove application adapters to develop integrations, without the need to know the technologies to build them.
We have also secure environments to access the tool together with a secure channel of communication for sending and receiving messages between our applications.

The given solution is integrate the Endeca Script with ICS and another script that gets the tweets (using the maximum id that we find directly in the last .json file, in order not to load more times the same tweets) before executing our graph.
This gives us the possibility to have an omogeneus application, that can give us the most recent tweets, and we can analyse them being able to understand also how the sentiment analysis evolves in the time.
The demo we have here is not the only case to apply this technology: we can apply it to several business case, with the use of Twitter (such Brand Analysis, Competitor Analysis, User Targeting, etc) and without it (Fraud Analysis, Business Analysis, etc).

Kontaktadresse:
Maria Emanuela Arico
esentri AG

Pforzheimerstraße, 132
D-76275 Ettlingen

Telefon:	+49 (0) 7243 / 354 90 0
Mobil:	+49 (0) 7243 / 354 90 63
E-Mail	emanuela.arico@esentri.com
Internet:	www.esentri.com