Transportation Research Analysis using NLP TechniqueS (TRANS)

Hyoungtae Cho, Melissa Egan, and Ferhan Ture

December 2, 2009

Abstract

Researchers in the transportation community are interested in analyzing the trends of research in their field. A website where people from the community could post their research needs, monitor and vote on others’ research ideas, view visualizations of research trends in the past, could be helpful to make this possible. In this paper, we describe our website project, titled Transportation Research Analysis using Natural Language Processing (NLP) TechniqueS, or TRANS. We explain the methodology we used to obtain data from the Transportation Research Board (TRB), extract information from data using NLP techniques, visualize the extracted information and publish a website that provides the required functionalities.

1 Introduction

People in the transportation research community are interested in understanding the trends and dynamics of their research field, collaborating with colleagues over research ideas in a convenient setting, and posting their research proposals to the community.

Researchers also have concerns about the ineffective use of research funds, and believe that repetitive research in the field is causing a waste of time and money. Many research ideas and projects are repeatedly published with slight repackaging. Another need in this community is the categorization of research projects, which is useful to track the amount of research done in each sub-field, understand research trends within the community, and as a result bring researchers with similar interests together.

At the moment, these tasks are partially managed by the Transportation Research Board (TRB), however this is both costly and ineffective for specific tasks. Performing the tasks computationally will produce fast, cheap, and qualitative results. Visualizing these results will make interpretation and analysis easier, and will communicate them to a larger portion of the community.

In this paper, we describe our analysis and visualization website project, titled TRANS. The goal of TRANS is to provide researchers with the tools they need to analyze research dynamics in their field, monitor others’ research need statements, and collaborate by voting on papers and research need statements. Natural Language Processing (NLP) techniques are used to automatically categorize research need statements and papers, discover repeated work, generate similarity lists in the website and therefore help people with similar interests get together. Information visualization methods are used to visualize the information using a scientific approach. All of
this is published on a website where researchers can submit their own research proposals, vote on others’ ideas and visualize various trends in the field.

This paper is organized as follows: Section 2 lists related work, and Section 3 describes the data we use, how it is acquired and preprocessed. In Section 4, we give a detailed explanation of the methods used, and Section 5 describes our visualization approaches with screenshots, and shows the final product: TRANS website. Some ideas for future work are included in Section 6, and finally Section 7 provides final remarks and concludes the paper.

2 Related work

The PopIT project is inspired by the assumption that, in any research area there are some concepts that are the main focus of researchers. The set of concepts that dominate the field change over time, along with the evolution of the field. This project aims to find how the actions and opinions of the individual researchers are related to the generation and change of core concepts in the field. The name comes from the fact that they focus on Information Technology as an example field, although they claim that the idea is applicable to any other field ([17, 19, 22, 17, 21]).

Science & Technology Innovation Concept Knowledge-base (STICK) provides a visualization to monitor and understand the innovations emerging in a research field. Three fields are used as examples in their work: information technology, biotechnology, and nanotechnology. The system uses NLP techniques in order to extract the necessary knowledge, such as the actors of innovations and relationships among innovations ([18, 20]).

Boyack and Börner propose a way to measure certain aspects of research, such as productivity of certain papers, amount of collaboration with other fields, trends and interconnections in the field, and determination of dynamic/static sub-fields ([6]). They also create visualizations to show interesting findings; one example is the relationship between amount of funding a certain paper or subfield receives and its productivity.

SCOPUS is a commercial tool that allows users to navigate through millions of research papers in its database. Some analysis functionalities are also available, but the main focus is to let the user search easily through papers in a database, rather than the tool inferring useful information from the data automatically ([2]).

There are many papers from a wide range of research areas that summarize the trends in research in the past years ([8, 13, 14]). This supports the idea that serves as a basis of this work: Researchers are interested in inspecting the general trends and patterns of research in their area, and it is not trivial to identify what these trends look like.

3 Data

Data is extracted from Transportation Research Board (TRB) web site using two different web crawlers, one to collect research need statements and the other to collect paper abstracts.

3.1 Research Need Statements (RNS)

The webpage (http://rns.trb.org) for RNS allows users to search a previous research need within 37 predefined categories and 3462 statements are currently available over last 4 years. We noticed that some of research statements are duplicated existing in different categories as well or regenerated with a few words changed, which is mentioned as one of motivations for this research project in the first meeting with our sponsor. After cleaning out the RNS dataset from the duplicates, we generate 809 data
instances, each consisting of title, description, date, sponsor committee, and category. In addition, we try to figure out proper classifications of research subjects with the cleaned data in order to replace the previous categories that are not well-balanced in terms of the number of statements contained. The category information of RNS data is used only for comparison with new categories that we create using NLP techniques.

3.2 Paper Abstracts

In order to compare research need statements with actual papers published before, we also crawl the web pages (http://trb.metapress.com/home/main.mpx) of TRB technical journal papers since 1996 and get their abstracts instead of the whole document. For these collected 9600 papers, the information of title, authors, and their addresses besides the content of abstract is also added so that we can show additional visualizations such as the research trends by state.

4 Methods

In this section, we describe the methods we used to infer useful information from the raw data. Our methodology can be explained in two parts: Clustering and pairwise similarity. For both parts, first we briefly describe the theoretical background, then state the motivation for using the method, and finally go into details on how we use them to solve our problem.

4.1 Clustering

Machine learning is the theoretical basis for mining useful information from raw data. There are two main categories of machine learning techniques: unsupervised learning and supervised learning. In supervised learning, the task is to learn a concept where both data points and labels are known. In unsupervised learning, the labels are not available, therefore the task is to find a representation of the underlying distribution ([9]). Recently, semi-supervised approaches have also become popular, where unsupervised learning is applied first (usually to a subset of the data) to assign initial labels, and then supervised learning is applied using the labels guessed in the first step ([3, 4]).

The data we have collected does not have reliable labels, therefore we focus on unsupervised learning and clustering in this paper. Due to space limits, our discussion of clustering is limited to its use in our methodology. For a more comprehensive overview of unsupervised learning and clustering, refer to papers [11], [12] and [5].

Clustering is the main tool for unsupervised learning, based on the simple idea of grouping similar data points together. Two aspects of clustering differentiate between various clustering algorithms: the definition of similarity between data points and the procedure of grouping the similar points together.

4.1.1 Distance measures

We can think of each data point as an \( m \)-dimensional vector, which we will refer to as feature vector for the rest of this paper. The whole data set is then a sequence of feature vectors, \( f^{(1)}, f^{(2)}, \ldots, f^{(n)} \). The similarity between two feature vectors can be defined using various distance measures. One of the most popular measures is Euclidean distance, where the distance between two feature vectors \( f^{(i)} \) and \( f^{(j)} \) is calculated as follows:

\[
    d_{Euc}(f^{(i)}, f^{(j)}) = \sqrt{\sum_{l=1}^{m} (f^{(i)}_l - f^{(j)}_l)^2}
\]
Measures that are widely used for clustering include Euclidean distance, Manhattan distance, maximum norm, Mahalanobis distance, mutual information, inner product and many others. Once the distance between two feature vectors can be computed, it is possible to group vectors that are similar to each other (high inter-cluster similarity) and not similar to other groups (low intra-cluster similarity). Every clustering algorithm attempts to find the ideal clustering that achieves high inter-cluster similarity and low intra-cluster similarity.

4.1.2 k-Means and Hierarchical clustering

Book chapters have been written about clustering methods and its applications, but we limit our discussion to two techniques and their application to the specific problem we want to solve.

The first technique is k-means clustering, a simple yet effective method that has been widely used for clustering. The algorithm starts with k randomly chosen ‘pivot’ data points, serving as cluster centers. Each data point is added to the cluster of which the pivot is closest to . The algorithm is done when each data point has been included in a cluster. The main advantage of this method is its low complexity (O(n)), however the quality of clustering depends on the initial k pivots and the number of clusters should be given by the user. There are variations where the number of clusters is discovered by the algorithm, but the best way is to test with several values and decide manually.

The second technique, or set of methods, is hierarchical clustering. In this case, the idea is to compute distance between all pairs of points, and merge the two that are closest. The same step is repeated with clusters and points, until there is one cluster that includes all points. When comparing distance between a cluster and a point, or more generally the distance between two clusters, there are three main ways. single linkage looks at the two closest points in the two clusters, complete linkage looks at the two farthest points in the two clusters, and average linkage looks at the average of distances between all pairs of points in the two clusters. The advantage of this method is the flexibility it gives to the user for choosing clusters, since the algorithm gives a tree (dendogram) as output. However, the time complexity is O(n^2), the algorithm is sensitive to outliers, and interpreting the tree is not always straightforward.

We use Weka machine learning toolkit ([10]) to perform k-means and hierarchical clustering on our data set. Weka is a simple Java-based toolkit that can be called from the command line, or a Java program, or using its own graphical interface. It contains many machine learning tools and methods, including the implementation of several clustering algorithms. It accepts input from a file format specific to Weka (called an arff file), but it is easy to convert an Excel sheet or delimited text file into their format. We experimented with the Simple k-Means implementation of k-means clustering, and the Cobweb implementation of hierarchical clustering.

4.1.3 Features

Due to time limitations, we only used unigrams as features, and we left our other ideas as future work in Section 6. For tokenization, the Galago tokenizer ([16]) was applied to clean text, remove stop words, and perform truncation. For instance, transport, transportation and transporting are all truncated to a prefix that is common to all words (e.g. transport). By doing this, words of same origin are grouped together, which makes the methodology more robust.

After preprocessing text with tokenization, a weight is calculated for each term in each document (research need statement or paper abstract) by applying the BM25 scoring function. The BM25 score of term t in document D is calculated as follows:
BM25\( (t, D) \) = \frac{\text{IDF}(t) \cdot \text{TF}(t, D) \cdot (k_1 + 1)}{\text{TF}(t, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\text{doclen}_{\text{avg}}})}

where \( \text{TF}(t, D) \) is the term frequency of \( t \) in the document \( D \), \( |D| \) is the length of the document \( D \) in words, and \( \text{doclen}_{\text{avg}} \) is the average document length in the entire collection. \( k_1 \) and \( b \) are free parameters, usually chosen as \( k_1 = 2.0 \) and \( b = 0.75 \). \( \text{IDF}(t) \) is the inverse document frequency of the term \( t \), computed using the following formula:

\[
\text{IDF}(t) = \log \frac{N - \text{DF}(t) + 0.5}{\text{DF}(t) + 0.5}
\]

where \( N \) is the total number of documents in the collection, and \( \text{DF}(t) \) is the number of documents containing \( t \) (document frequency).

### 4.1.4 Reducing dimensionality

This approach yields 13514 unigrams, therefore the dimension of each feature vector is 13514. This creates problems for computationally complex algorithms like Cobweb hierarchical clustering. A typical approach to reduce the dimension is to remove very common and very rare words, because they contain less information than other words. To do this, we impose a lower and upper limit on the document frequency of words to be included as features. After experimenting with different values, we used a lower limit of 5 and upper limit of 400 on the research need statements’ collection of 809 documents. This reduced the number of features to 2576.

### 4.1.5 Category assignment

When we apply a clustering algorithm to our data, it assigns each data point to a cluster, but the clusters do not have labels or meanings. For instance, the first instance is assigned to \( \text{cluster}_2 \), the second is assigned to \( \text{cluster}_0 \), and so on. Since we are interested in labeling each data point with a category, we need a way to map clusters to categories.

Our solution is to get an expert of the domain to do this assignment manually, for the first time, and then use those categories to perform future assignment automatically. We assume that an expert can easily come up with category names after looking at sample text and most frequent words of a cluster. We must emphasize that this manual assignment must be done only once to the initial data. Later, any research need statement \( x \) can be automatically labeled by finding the cluster with the centroid closest to \( x \), and choosing the category associated with that cluster. If the number of clusters is \( k \), and the distance measure is \( d \), below is the formula to assign category to document \( x \):

\[
\text{CLUSTER}(x) = \arg\min_{i=1}^{k} d(\text{centroid}(\text{cluster}_i), x)
\]

\[
\text{CATEGORY}(x) = \text{category assoc. with } \text{CLUSTER}(x)
\]

For hierarchical clustering, the same one-time manual assessment needs to be done for deciding on the cluster cutoff points, looking at the dendogram. We understand that this may be problematic in some cases and may not be feasible for replication, but this is our best alternative due to time constraints of this project. Other approaches are discussed as future work in Section 6.

### 4.2 Pairwise similarity

Pairwise similarity computation is an important problem in information retrieval that deals with finding the similar document pairs in a collection of documents. It has been used in many applications such as similarity list generation, clustering objects, unsupervised learning, and retrieval of various information from a collection. As explained above, let’s as-
sume that each document is represented by a feature vector containing term weights. The pairwise similarity score based on this model is the cosine of the angle between the two vectors in a collection:

\[ \text{PWSIM}(u, v) = \cos(\theta(u, v)) = \frac{|u \cdot v|}{|u||v|} \]

where \( u \) and \( v \) are feature vectors representing the documents, and \( \theta \) is the angle between these vectors. The goal is to find the top \( N \) documents that have the highest similarity score with a given document.

As the number of terms increase, the dimension of feature vectors increases, and the time complexity of tasks like clustering or pairwise similarity increases quadratically. High dimensionality is an important problem in many tasks involving operations on feature vectors.

### 4.2.1 Generating signatures

Ravindran et al describe how to use locality sensitive hash functions to reduce the dimension dramatically, while keeping accuracy high ([15]). In this approach, \( D \) random vectors are used to map each document vector \( u \) from the domain \( \mathbb{R}^k \) to a “signature” \( s \) from a much smaller domain \( [0, 1]^D \). Each bit of \( s \) is determined by a dot product of \( u \) and the corresponding random vector. Given \( D \) random vectors \( r_1, \ldots, r_D \),

\[ s_i = 1 \text{ if } u \cdot r_i \geq 0 \]
\[ 0 \text{ otherwise} \]

Using these signatures, the cosine similarity score between two documents can be calculated by simply counting the number of bits that differ. This count is called the hamming distance, and the cosine similarity score is calculated by the following formula:

\[ \text{PWSIM}(u, v) = \cos\left(\frac{\pi \text{(hamming distance)}}{D}\right) \]

### 4.2.2 Computing similarity scores

Because each document is represented by a fixed-length sequence of bits, it is possible to compare documents from different collections. We execute the following procedure to compute signatures:

1. For document \( d \), we compute the feature vector \( f \) as explained in Section 4.1.3.
2. An additional step is to normalize the vectors so that the norm of each is 1.
3. Given the number of bits \( D \), we randomly generate \( D \) unit vectors.
4. Each bit of the signature is computed by the dot product of a random vector and \( f \).
5. Steps 1–4 are repeated for each document (research need statement or paper abstract) in the collection.

After generating signatures for all documents in the collection, the pairwise similarity score between any two documents is easy to calculate.

We can use this method for two purposes: First, applying this to all pairs of documents and sorting the results gives us the most similar pairs of documents. This can reveal interesting information such as repeated research, by finding very similar research statements. Cross-similarity between research need statements and paper abstracts may also reveal interesting correlations. The second purpose is to have a website feature where a more-like-this list is generated when the user submits a research need statement. The list may be limited to paper abstracts, since showing previous papers that match the user’s research needs is useful.
5 Visualization and Website

5.1 Prefuse

In order to create a dynamic and interactive visualization and meet the needs of users, we use the Prefuse toolkit ([1]) that provides a set of information visualization frameworks implemented in Java. Among alternatives, our highest priority is to show the graph of trends in the transportation research over years, in which the user will interactively find insights, such as (1) most popular issues of the year, (2) topics that are promising or losing interests, and (3) topics that have drastically or hardly changed over years.

5.1.1 Data structure

The data structure we used for our visualization is a graph structure consisting of nodes and edges in the GraphML format ([7]). We use tree nodes to represent the popularity of a category in a certain year and tree edges to show the change of popularity for a category between two consecutive years. Each node has five elements as following:

- Category name
- X and Y coordinates that indicate its location
- A value indicating the color of a node
- Year

while each edge has two elements:

- Name of edge composed of Category name and from/to years
- A value indicating the color and thickness of the edge

5.1.2 Graph Panel

The interface creates a renderer factory based on the constructed data structure. As shown in Fig. 1, labels on the left rank the categories based on the percentage of research statements in each category in the starting year, and labels on the right show the corresponding ranking for the last year. X-axis at the bottom of the graph indicates years that the statements have been submitted, and they can be filtered on the control panel. We code the color and thickness of edges in graph by assigning high values to lines that oscillate a lot using the following formula:

\[
value(c) = \sum_{i=1}^{n-1} |\log \frac{rank_{i+1}(c)}{rank_{i}(c)}|
\]

where \(rank_{i}(c)\) is the rank of category \(c\) in \(i^{th}\) year and assuming there are \(n\) years. Notice that the value is higher as oscillation increases, and gets closer to 0 as the graph gets closer to constant.

![Figure 1: the graph visualization of research trends using Prefuse](image)
5.1.3 Control Panel

On the right pane as a control panel, we have several tools such as filter with dynamic visualizations. For example, categories of the paper are lexicographically ordered by their name and then use a suite of double-edged sliders to filter out papers that are not interesting to the user.

5.1.4 Functionality

Once the mouse is over an edge of the graph, the edge and its neighbored edges are highlighted in blue and the label bottom left shows the name of edge as shown in the Fig. 2. The functionality for mouse-over event helps to recognize which edge is currently selected.

In addition, we provide a search box to find the name of category and dynamically highlight matched nodes in blue on the graph, with count shown next to the input box. In Fig. 3, when 'p' is entered in the search box, two lines representing 14 matched results of nodes and edges are colored with blue since their category names are starting with 'p' such as 'performance-testing2007' or 'public-transit2009'.

5.2 Website

Our website was implemented using a three-tier architecture. At the presentation level we have hypertext markup (HTML) pages that may be viewed from the user’s web browser of choice. Our website is platform-independent, although its presentation has been optimized for Firefox. At the application logic level we make use of an Apache web server and a Perl CGI (Common Gateway Interface) script. These components work together to generate HTML documents and serve those HTML documents to the user. They also process user arguments submitted via HTTP GET and HTTP POST.
commands and generate customized HTML pages in response to these user commands. At the back end, the CGI script loads and stores data from flat text files. The current web site implementation is a prototype only; were it to be adopted for professional use, data management would ideally be accomplished using a full-scale database management system such as MySQL or Oracle. With this in mind, the portions of the CGI script that involve loading or storing data have been modularized so that they may be easily replaced.

The website allows access to our Prefuse graph visualization, but also provides several additional functions:

- the listing, sorting, and searching of research needs statements
- the ability to identify research needs statements most similar to a statement of interest
- the ability to submit new research statements
- the ability to vote on existing research needs statements
- the ability to discover users with similar interests based on their voting records and profiles.

The website also provides an alternative JavaScript-based visualization that depicts rising and falling topics among research needs statements over the years. These topics were determined using latent Dirichlet allocation (LDA), an unsupervised topic modeling algorithm that characterizes topics as distributions over words and documents as distributions over topics. Because the algorithm is unsupervised, topics are not automatically assigned labels; in the visualization, each topic is represented by a list of its most probable constituent words. When a user mouses over a given topic, that topic is highlighted within the ranked list of topics corresponding to each year. This allows the user to easily identify trends in the popularity of that topic over time. Additionally, topics are color-coded using a gradated color scheme corresponding to their rank within a particular year of interest, so topics whose ranks have risen or fallen drastically since that year are easy to identify at a glance. Further potential refinements to the visualization are described in the section on future work.

Beyond the standard searching, sorting, and filtering functions common to web-enabled databases, our website applies the NLP techniques described earlier to provide additional useful functions to users. One such function is the automatic recommendation of research statements similar to the one the user has selected, based on pairwise similarity scores. Not only will this alert the user to statements that may be interesting and relevant to his or her subfield, but it will allow for the automatic detection of duplicate or near-duplicate research statements.

When a user submits a new statement, he or she is likewise presented with a list of similar pre-existing statements in the course of the submission process.
This allows for corrections and refinements to be made based on similar research, and helps to preemptively eliminate the problem of duplicate or near-duplicate submissions. To promote consistent categorization of research statements, the system automatically proposes categories for newly submitted statements by making use of the clustering results described earlier. Users are free to modify these categories if desired.

The web site provides a mechanism to vote for or against specific research statements, and users may view and sort research statements by their approval rate among web site users in general. To encourage active participation in the voting process, approval rates are only visible after a user has voted or specifically declined to vote on a particular statement. This restriction is also intended to prevent prior approval rates from biasing a user’s own vote. A voter may at any time access his or her own voting record, which includes statistics about his or her voting patterns over various periods of time. As one goal of the project was to allow users to discover others with similar interests, the voting record page also includes a list of other users with the most similar voting records. These records may be examined side-by-side with differences highlighted, and user profiles may be consulted to determine the fields of interest and contact information of similar users.

6 Future work

6.1 Other Visualizations
We have implemented a temporal graph of trends in transportation research. Due to time constraints, other visualizations requested by our sponsor are left for future work.

1. **Bar chart or Treemap** to show the number of research need statements for each category, and if the user clicks on the particular category, then the chart will be expanded into its sub-categories

2. **Map visualization** for the research trends in details by state or by region

3. **Stacked graph** to represent overall research trends over years

6.2 Better Features

6.2.1 Use higher order n-gram features
In this project, unigram features are used to measure the similarity between documents and to find clus-
terings. Although using only unigrams seems to be sufficient, we have a hunch that higher order n-gram models can provide better features to consider frequently used word phrases such as public transportation.

6.2.2 Use transportation ontology

All the words are normalized to group words of the same origin together for the robust feature vector. Despite the normalization, the dimension of feature vector is so huge to dramatically increase the complexity in running hierarchical algorithm. The analysis of transportation ontology instead of removing rare words would reduce the vector space by normalizing all similar words with regards to transportation terminology, so that we could preserve the information gained from rare words when computing similarity.

6.3 LDA topic presentation

The collections of words associated with unsupervised LDA topics may not necessarily be intuitive or meaningful to users. Work could be done to better communicate the meaning of the topics or how they were derived, and to assign meaningful and concise labels to these topics. It could also be informative to make use of multi-word phrases, and not just individual words, in the topic-modeling process. A transportation ontology could be consulted to determine meaningful phrases within the field.

6.4 Citation network analysis

A major focus of this project was the consistent categorization of documents and the automated discovery of similar documents, and citation network analysis could have provided us with an additional angle from which to attack this problem. Citation network exploration techniques such as those implemented within the iOpener workbench could have revealed communities of similar documents, as determined by the papers that they cite and the papers that cite them. This avenue of exploration is left for future work.

7 Conclusions

We have implemented a dynamic Java-based visualization to illuminate the trends within transportation research over the years. We have also implemented a web site to assist users in the discovery, submission, and evaluation of transportation-related statements of research. We believe that our work on this project can be used to make trends within transportation research more transparent to those within the research community, and to promote collaboration and wise funding decisions within the community.

Acknowledgements

We would like to thank Michael Pack and Prof. Ben Shneiderman for their continued support and guidance throughout this project.

References


