FUNG INSTITUTE
for ENGINEERING LEADERSHIP
UC BERKELEY COLLEGE OF ENGINEERING

# Visualizing Key Words and Trends in a Subset of the US Patent Database

Andrew B. Godbehere
Laurent El Ghaoui
David Liu
Sumeet Sharma

College of Engineering
University of California, Berkeley

Fung Technical Report No. 2014.05.07
http://www.funginstitute.berkeley.edu/sites/default/files/Visualizing

May 7, 2014

The Coleman Fung Institute for Engineering Leadership, launched in January 2010, prepares engineers and scientists – from students to seasoned professionals – with the multidisciplinary skills to lead enterprises of all scales, in industry, government and the nonprofit sector.

Headquartered in UC Berkeley's College of Engineering , the Fung Institute combines leadership coursework in technology innovation and management with intensive study in an area of industry specialization. This integrated knowledge cultivates leaders who can make insightful decisions with the confidence that comes from a synthesized understanding of technological, marketplace and operational implications.





130 Blum Hall #5580 Berkeley, CA 94720-5580 | (510) 664-4337 | www.funginstitute.berkeley.edu

**Abstract:** We describe the design and implementation of an interactive content visualization tool as applied to a subset of the US Patent Database pertaining to clean technology. The tool determines the "image" of a keyword query by selecting a few words most strongly associated with the query. The process is repeated for several distinct time windows, yielding a time evolution of the relative importance of given words in association with the query. Further, with each identified word is an associated list of patents from the given time window mentioning both the query terms and the resulting output word, allowing a user to browse relevant examples. As we open the tool to the public, we aim to provide new insights into the overall trends and relationships in innovation.

# Visualizing Key Words and Trends in a Subset of the US Patent Database

Andrew B. Godbehere, Laurent El Ghaoui, David Liu, Sumeet Sharma

May 7, 2014

**Abstract**

We describe the design and implementation of an interactive content visualization tool as applied to a subset of the US Patent Database pertaining to clean technology. The tool determines the "image" of a keyword query by selecting a few words most strongly associated with the query. The process is repeated for several distinct time windows, yielding a time evolution of the relative importance of given words in association with the query. Further, with each identified word is an associated list of patents from the given time window mentioning both the query terms and the resulting output word, allowing a user to browse relevant examples. As we open the tool to the public, we aim to provide new insights into the overall trends and relationships in innovation.

## 1 Introduction

The visualization we employ is used in StatNews (`www.statnews.org`), an ongoing research project of Professor Laurent El Ghaoui at the University of California at Berkeley. StatNews was developed specifically to explore the image of topics in traditional news media coverage. We modified the computational framework to accommodate a subset of the US Patent Database, centered on clean technology. The details of the dataset are described in Section 3.1.

The visualization and underlying computation were first described by Gawalt et. al [3]. This view is referred to as a "staircase" in reference to its characteristic appearance. Gawalt et. al [3], focused on application of the technology to traditional news media. To apply the approach to patent data, each of which is much larger than a typical news article, required careful data structure design for efficient manipulation of large datasets.

The tool we provide interacts with a database of roughly 30,000 patents with application dates between May 01 1957 and July 09 2013. All patents have been identified as being relevant to the topic of "clean technology," as developed by Li et. al. [6]. Given the large size of every document, it would be impossible to read and comprehend the contents of every patent, let alone identify interesting trends. We employ machine learning tools in order to provide succinct analysis of a comprehensive analysis of the text.

## 2 Query Functionality

### 2.1 Specifying a Query

Queries are comprised of a boolean combination of text terms, a desired time range, and the time granularity level for the search results.

#### 2.1.1 Text Query

The text portion of the query can be specified by three lists of terms. The first is "ALL," indicating that results matching your query must mention ALL of the words in this part of the query. The second is "ANY,"

indicating that results matching your query must mention at least one of the query terms in this part of the query. The last is "NONE," which means that matching results must not mention any of the query terms listed in this section.

As an example, suppose the query is :

- ALL: "wind", "rotor"
- ANY: "gear", "power", "motor"
- NONE: "water", "wave"

The complete Boolean combination of this query is: "(wind AND rotor) AND (gear OR power OR motor) AND (NOT(water OR wave))". Matching documents must mention both "wind" and "rotor", any of "gear", "power", or "motor", and neither "water" nor "wave".

### 2.1.2 Time Range

To investigate the time evolution of the image of these text queries, a date range must be specified. The time range is broken down into a sequence of time windows, within which independent analyses are run in order to present a time evolution in the results. We permit scanning through time in three different granularities, by month, quarter, or year. Months are defined to begin on day 1, quarters are defined to start on the first of *Jan*, *Apr*, *Jul*, and *Oct*, representing 3-month intervals. Years all begin on the first of the year.

## 2.2 Interacting with the Tool

The tool is available for use at `statnews.org`. Selecting "Clean Technology Patents" as the News Source will direct the tool to analyze the clean technology patent database described in Section 3.1.

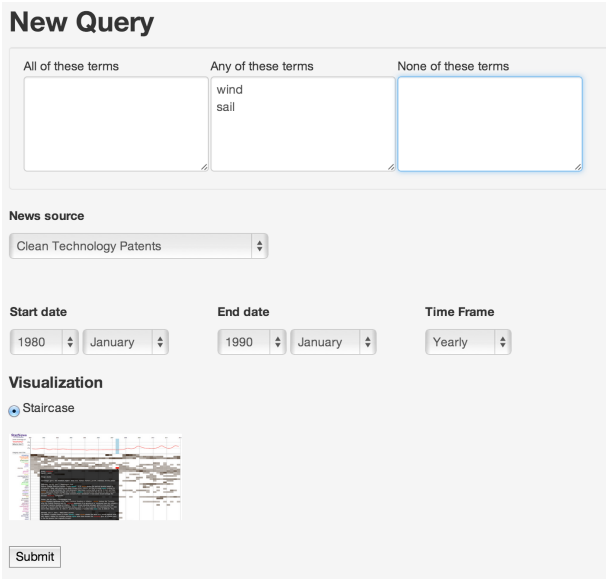


Figure 1: Query Interface Example

As shown in Figure 1, the first step is to specify the text query. Three text boxes allow entry of one query term per line. The boxes represent search fields for "ALL", "ANY", and "NONE", as described in Section 2.1.1. In the example above, the query is searching for documents mentioning one or both of the words "wind" and "sail".

News Source must be set to "Clean Technology Patents" to query the patent database. StatNews provides interfaces to other data sources, such as a collection of articles from the New York Times.

The third step is to specify a time range for the query, which are set by drop-down selections for year and month. The time frame drop-down determines the width of each analysis interval. In the example above, each distinct year is analyzed independently; other options are to analyze by quarter or month.

When the query is fully specified, press "Submit" to process the result. In a few moments, the user's window will refresh with the results.

## 2.3 Interpreting the Result

An example staircase visualization is presented below:

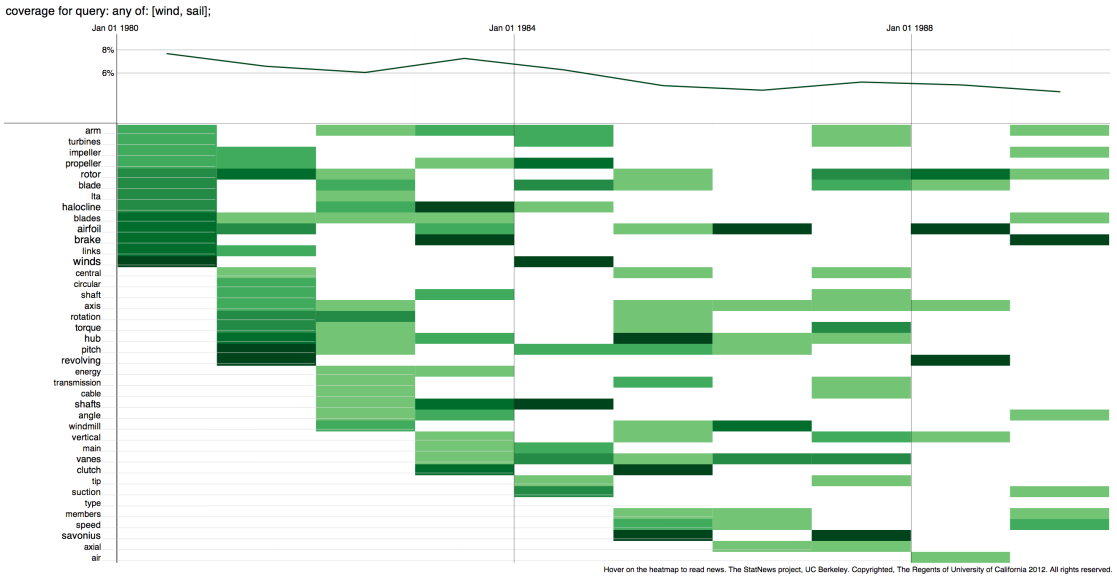


Figure 2: Searching for either "wind" or "sail" yearly in the 1980's.

First, the top portion of the graph depicts a line-graph. This line-graph indicates how much of the overall dataset matches the query. In this case, the overall fraction of patents mentioning "wind" or "sail" begins at nearly 8% at the beginning of the decade, and declines steadily to about 5% by the end of the decade. This yields one insight into the temporal trend of the queried topic, similar to what is presented with the Google n-gram viewer [7], available online at `https://books.google.com/ngrams/info`. This does not necessarily indicate that the number of new patents relating to "wind" or "sail" technology declined; rather, it indicates that they represent a decreasing proportion of the overall number of new clean technology patents throughout the 1980's.

Along the left axis of the graph is a list of terms that appear, representing the "image" of the query. These are words that are most associated with the query as compared to patents not matching the query. The words are sorted vertically in order of appearance over time, yielding the staircase appearance of the shaded rectangles. Another quick insight to be gained from the chart is that the list of words on the left are the words most associated with the query across the entire time range. In the 1980's, for example, many of the terms related to "wind" or "sail" involve mechanisms such as "turbines", "rotors", and "vanes", along with actions such as "rotation" and "revolving".

The shaded rectangles are grouped into ten vertical "slices" each representing the results of one year of the overall query span (one decade). The darkness of each box represents the weight given to its corresponding term in the results relative to the other words. Having no box indicates that the given word did not appear as one of the most significant words corresponding to the query in the given time interval. Reading the

graph horizontally illustrates individual words becoming more or less important to the query. For instance, the word "blades" is marked with reducing importance from 1980 until 1983, at which point it no longer appears to be particularly associated with the query. The importance of "hub" varies quite a lot over time.

Clicking on any shaded rectangle brings up a list of one or more patents that match the query and mention the word selected. The title of the patent is highlighted, and presented on the following line are the patent number, application date, and inventor's name, followed by the patent abstract. This view helps elaborate on the keyword results by providing explicit references to relevant patents. For example, in Figure 3, the term "savonius" is explained in 1985 by calling up a relevant patent, in this case describing an improvement in the efficiency of a variety of wind turbine. Such focusing-in on results can offer more context to the single terms in the staircase results. This feature also presents a list of patents relevant to the original search term, and distributes them according to keyword, which presents an interesting non-linear browsing methodology in studying patent trends.

For more on the staircase graph, we refer the reader to the excellent outline by Gawalt et. al. [3].


Figure 3: Clicking on "savonius" in 1985

# 3 System Architecture

In this section, we describe the database configuration, pre-processing, and computational pipeline to prepare the visualization to meet a user's query. All software used is written in Python 2.7 [2], available from `www.python.org`. We use numerical computation routines from python libraries Numpy and Scipy [5]. Machine learning algorithms are imported from the open source library scikit-learn [8].

## 3.1 The Dataset

The segment of data we work with is selected to pertain to clean technology. We store the data in a single SQLite database table. The dataset incorporates 29,447 patents with application dates between May 01 1957 and July 09 2013. The dataset was developed by Li et. al. [6]. The text content utilizes 601,420 distinct words, and with each query, our goal is to identify only approximately 20 that are best associated with the query. The database contains the following information on each patent:

- Patent Number

- Application Number
- Application Date
- Date Granted
- Inventors
- Assignee
- US and International Classes
- *Title*
- *Abstract*
- *Claims*
- *Description*

The italicized fields contain the text content that will be analyzed. All dates are stored in the database not as text but as integers, in Unix Timestamp, represented as the number of seconds elapsed since the epoch, Jan 01 1970, and negative numbers for dates before the epoch. This enables the fast time-window searching within the database that is necessary for our application.

### 3.2 Preprocessing

To utilize numerical machine learning algorithms efficiently on large bodies of text, we first represent the text in a bag-of-words model, as described in [3]. In brief, each document is represented as an unordered collection of words (sometimes referred to as features, in accordance with other machine learning literature). Capitalization, punctuation, and word ordering are ignored, and a patent is represented by the set of words that appear in the document, and how frequently those words appear.

The machine learning algorithms we use to create the visualization operate on a document-by-term matrix, which in this case is of dimension $29447 \times 601420$. Entry $i, j$ corresponds to the number of times word $j$ appears in document $i$. It should be noted that the vast majority of these entries will be 0, as each document only mentions a small fraction of the words comprising the entire lexicon, and the matrix is sparse. This enables us to represent and manipulate the data compactly, an important feature for a real-time computational system.

The first stage of preprocessing, then, is to identify valid words and represent the document as a row in a matrix. We use regular expressions [4] to identify valid and invalid tokens. First, regular expressions are used to eliminate would-be words that are either numbers or short words (comprised of only one or two characters). Second, all URLs and other non-prose markup is removed. Then, all remaining words (alphabetical character sequences separated by whitespace) are collected and counted. At this point, the text has been "vectorized," as each document is represented as a collection of numbers.

To quickly process the monthly, quarterly, or yearly time slices, we create each of the sparse matrices for each time slice in advance, and save them in a special compressed format on disk, which is very fast to load into memory.

## 4 Data Analysis

As described in Gawalt et. al. [3], the words identified in each time interval are chosen via a process called "feature selection," in general. To achieve fast on-line computation, we use modified linear regression models: sparse logistic regression [3] or a problem called the Lasso [9]. Each are numerical optimization approaches designed to select only a few of the overall set of features, so the result may be easily read and interpreted by a human.

Upon receipt of a query, documents are grouped into time windows and then labeled according to whether or not they match the query. A classification vector $y$ of the same length of the number of documents within a given time window is created, and patents matching the query are labeled with a $+1$ and otherwise with a $-1$. The result creates a vector of weights $w$, one for each word in the dataset. By nature of the sparse algorithms used, most entries of $w$ are 0. The few largest weights identify the words to be used for the visualization.

To refine the results, we employ TF-IDF normalization, a standard practice in text analytics [1]. Aizawa et. al. [1] describe a feature-selection methodology designed to "prune infrequent words", "prune high frequency words", and to choose words "which have high mutual information with the target concept."

# References


[1] Aizawa, A. An information-theoretic perspective of tf-idf measures. *Information Processing and Management 39* (2003), 45–65.

[2] Foundation, P. S. Python, 2014.

[3] Gawalt, B., Jia, J., Miratrix, L., El Ghaoui, L., Yu, B., and Clavier, S. Discovering word associations in news media via feature selection and sparse classification. In *Proceedings of the international conference on Multimedia information retrieval* (2010), ACM, pp. 211–220.

[4] Goyvaerts, J. Regular Expressions Reference, 2014.

[5] Jones, E., Oliphant, T., Peterson, P., and Others. SciPy: Open source scientific tools for Python.

[6] Li, G.-C., Paisner, K., and Fleming, L. A List of Clean Tech Patents. Tech. rep., Fung Institute, University of California, Berkeley, 2014.

[7] Michel, J.-B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., Pickett, J. P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M. A., and Aiden, E. L. Quantitative analysis of culture using millions of digitized books. *Science (New York, N.Y.) 331* (2011), 176–182.

[8] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[9] Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological) 58*, 1 (1996), 267–288.