Preliminary Report on Early Growth Technology Analysis

Blaine Ziegler, Ayse Kaya Firat, Clare Li,
Stuart Madnick, Wei Lee Woon

Working Paper CISL# 2009-04

April 2009

Composite Information Systems Laboratory (CISL)
Sloan School of Management, Room E53-320
Massachusetts Institute of Technology
Cambridge, MA 02142

Preliminary Report on
Early Growth Technology Analysis

Blaine Ziegler
Ayse Kaya Firat
Clare Li
Stuart Madnick
Wei Lee Woon

24 June 2009

ABSTRACT

Even an expert cannot be expected to be fully aware of all new and promising technology developments in broad complex fields, such as renewable energy. Fortunately, there are many diverse sources of information available, such as journal publications, news stories, and blogs, which are growing at enormous rates. But it is not realistic that any human could read and digest the contents of all this information – especially in a timely manner.

In this paper we describe how we can apply technology mining techniques to these diverse information sources to study, visualize and identify the evolution of promising new technologies – what we refer to as "early growth technology analysis." For the work reported herein, we use as sources the information about millions of published documents contained in the SCIRCUS, Inspec, and Compendix sources.

We accomplish our analysis through the use of bibliometric data mining analysis, consisting of three key steps:

1. Identify related keywords (from keywords in articles)
2. Determine frequency of occurrence per year of these keywords
3. Determine those with rapid growth from a low base

In this paper, we use the topic "renewable energy" as our case study example.

1. Introduction to the concept of Early Growth Technology Analysis

The goal of this project is to identify, in an automated manner, potentially important technologies that have not yet become mainstream and well-known. We refer to such technologies as “early growth” technologies. One way to view such technologies comes from the consideration of a plot of a technology’s “prevalence” over time. We are interested in those technologies that are in the early, low-prevalence tail of such a plot as depicted in Figure 1.

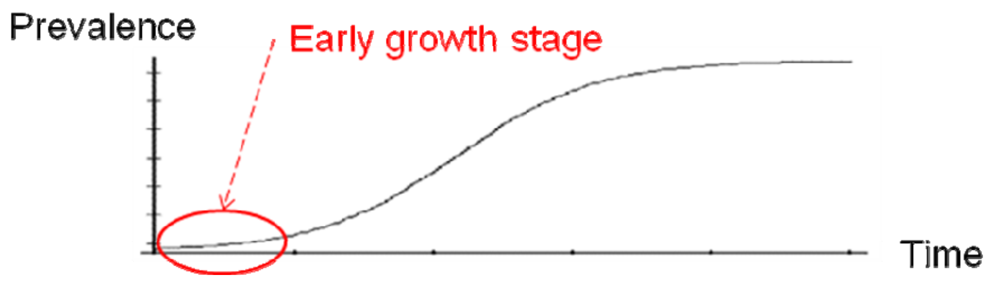


Figure 1. Prevalence of Early Stage Growth over Time

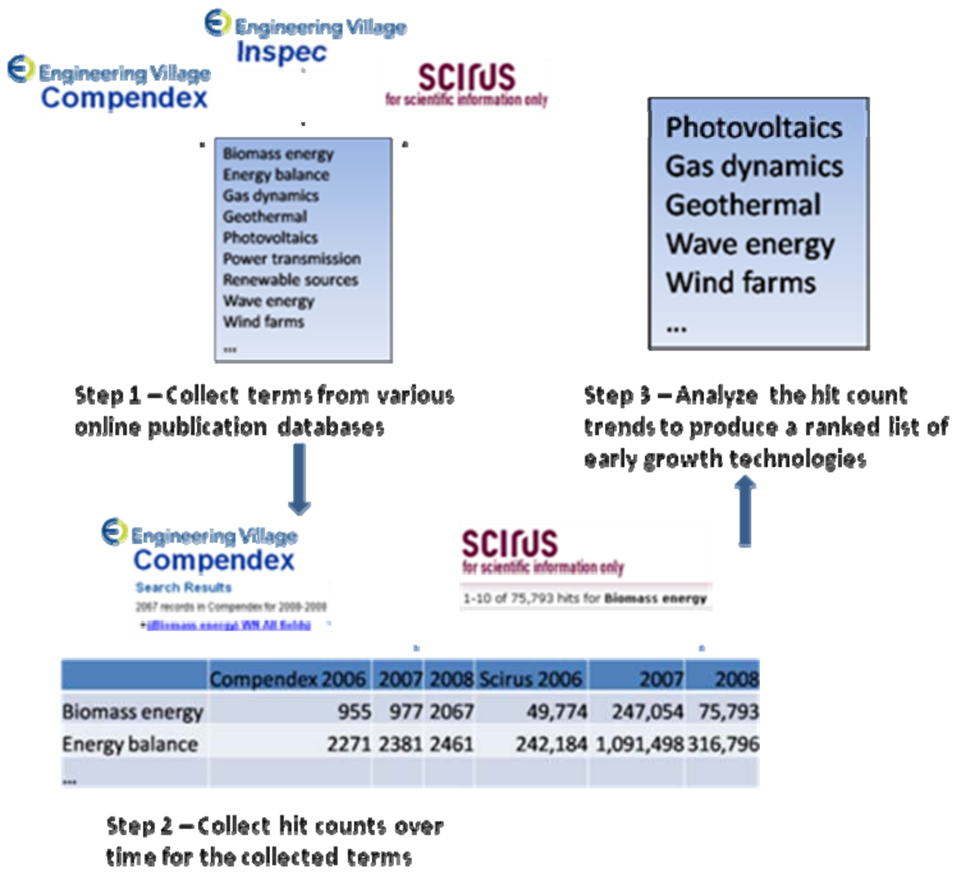
An immediate issue is what do we mean by “prevalence,” and even more importantly, how could we measure it? In this project, we use bibliometric analysis. That is, we look for the frequency of mentions of a technology over time in the published literature[1].

We aim to be able to forecast any of such technologies, but to make the research more concrete, we are using renewable energy technologies as a case study. This is a good topic because it is expected that many new technologies will emerge in this field in the upcoming months or years.

2. Our Basic Methodology

Our methodology consists of three steps. First, we collect a large number of terms related in some way to the broad technology field of interest, renewable energy in our case. Second, we record the change over time of hit counts from online publication databases for each of our collected terms. Third, by studying the growth pattern of the hit counts of these terms over time, we produce a list of technologies that are in the early stages of their growth. For future work, we plan to perform a second pass on this list with more rigorous, computationally intensive analysis to further reduce our list to a small number of important terms. Each of these steps is elaborated in this report. Figure 2 below give a high-level representation of the methodology.

[1] We are also exploring sources such as news articles and blogs, but we do not discuss those efforts in this report.


Engineering Village
Inspec
Engineering Village
Compendex
SCIRUS
for scientific information only
Biomass energy
Energy balance
Gas dynamics
Geothermal
Photovoltaics
Power transmission
Renewable sources
Wave energy
Wind farms
...
Photovoltaics
Gas dynamics
Geothermal
Wave energy
Wind farms
...
Step 1 – Collect terms from various online publication databases
Step 3 – Analyze the hit count trends to produce a ranked list of early growth technologies
Engineering Village
Compendex
Search Results
2067 records in Compendex for 2008-2008
+((Biomass energy) WN All fields)
SCIRUS
for scientific information only
1-10 of 75,793 hits for Biomass energy
Step 2 – Collect hit counts over time for the collected terms

	Compendex 2006	2007	2008	Scirus 2006	2007	2008
Biomass energy	955	977	2067	49,774	247,054	75,793
Energy balance	2271	2381	2461	242,184	1,091,498	316,796
...						

Figure 2. Three steps used in Methodology

2.1 Step 1 - Collection of relevant terms

To begin our search for interesting early growth technologies, we must first construct a long list of terms related to our topic of interest, renewable energy. At this preliminary step, we are not particularly concerned with screening the terms for quality; we are only interested in gathering a large number of terms somehow relevant to the topic. For our initial experimentation, we have chosen to use a few thousand terms.

The key sources that we use are online publication database systems (see Appendix 1 for descriptions and discussions of some of the sources.) Examples include Compendex, Inspec, Scirus, and Google Scholar. It is important to note that these are web-based services. We do not have, nor do we need, physical access to the actual contents of these databases. We can ask a query, such as “What papers do you have about ‘renewable energy’” and be provided with a list of such papers or ask “How many papers do you have about ‘renewable energy’” and be provided with that number – often called the “hit count.”

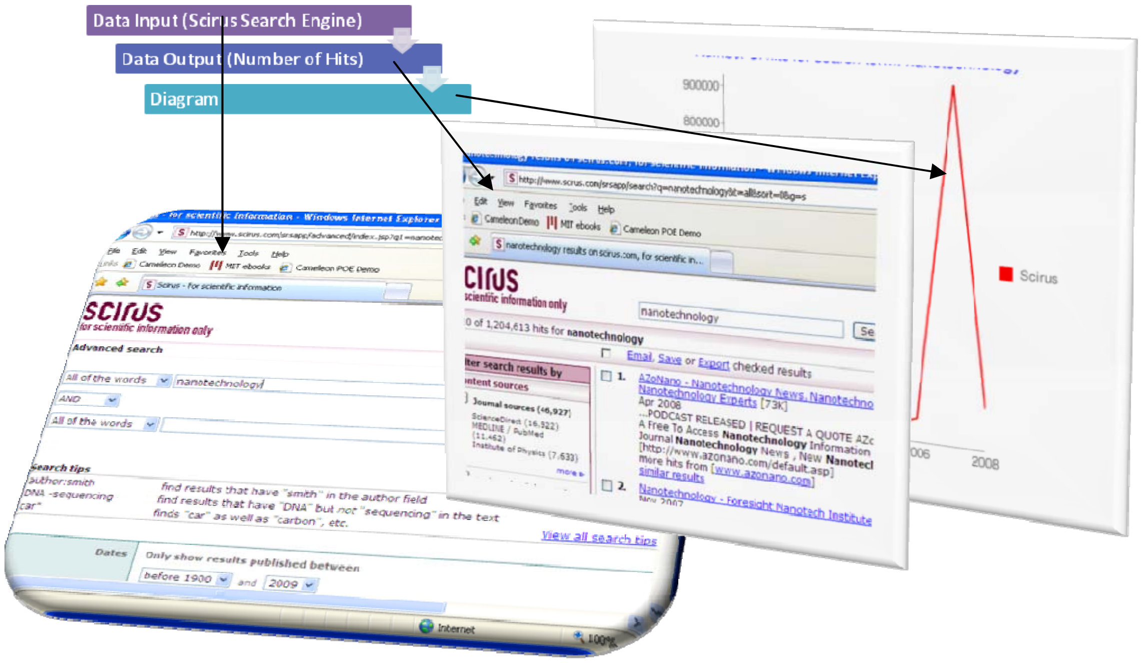

Data Input (Scirus Search Engine)
Data Output (Number of Hits)
Diagram
Scirus

Figure 3. Illustration of Retrieving "hit count" from Scirus source

As with all four steps in this project, we aim to construct this list of terms in a completely automated fashion. Specifically, we run software that uses these online publication databases to locate and extract terms that are relevant to the topic of interest. The particular method for this automated term collection necessarily varies depending on which publication databases are being used due to the differences in the way that these database systems operate. In fact, it does not appear to be possible to perform such term collection with all existing databases. An open area of ongoing investigation involves finding the best databases to use for this step. Important criteria to consider are the databases' size and focus, as well as the feasibility of using them for term collection. For now, we have found that the following three sources work well: Compendex, Inspec, and Scirus[2].

2.1.1 Term collection using Compendex and Inspec

Compendex and Inspec both fall under the umbrella website Engineering Village. We can use them to collect terms in the following way:

1. We first use their search engine to search for articles about the topic of interest. A list of those articles is returned sorted in order of relevance. (The precise definition of relevance is unclear, but for now we assume that the sorting matches our intuitive understanding of relevance.)
2. For each article identified, a link is provided to its abstract. When the abstract is viewed, along with the abstract text, there is also a list of keywords.
3. These keywords were provided by the author, who uses the terms to indicate the article's topic(s). These author keywords come in two varieties, namely controlled and uncontrolled keywords. Controlled keywords come from a list of thousands of terms, from which the author can choose those that are relevant. Uncontrolled keywords, on the other hand, are entered by

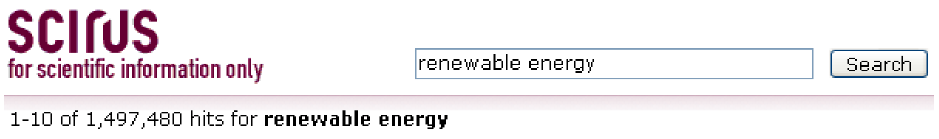
[2] See the Appendix 1 for more detailed information about these systems.

the author and are not restricted to the premade list. We currently do not put any more or less value on either set of keywords; the controlled and uncontrolled terms are combined into one list.

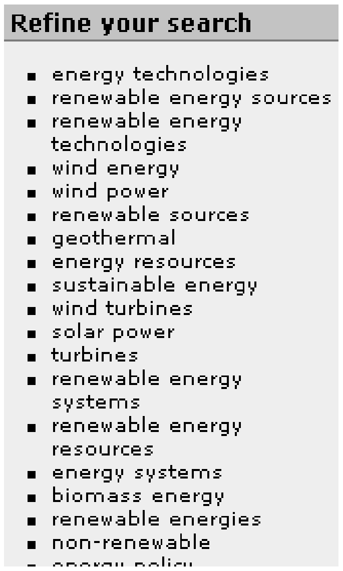
To use Compendex or Inspec to collect keywords, our software first queries the database for the topic, "renewable energy," visits the abstracts of the most relevant articles, and collects all the keywords listed with each of these abstracts. Duplicate terms are removed. In keeping with our goal of collecting a few thousand terms, we are currently using the 70 most relevant articles, but the number of articles is a tunable parameter. When using 70 articles, Compendex returns approximately 1,600 unique terms, while Inspec returns about 1,400.

2.1.2 Term collection using Scirus

Scirus is the third data source that we have used to collect terms. Due to the nature of the Scirus, the collection method is entirely different from that which we use for Compendex and Inspec. Scirus does not provide keywords associated with the articles it returns. But it does provide a "Refine your search" feature for given a search query, as shown in Figure 4, which displays a list of 80 related terms. (In some cases, the list may contain fewer than 80 terms, if the search query is quite obscure. The list never contains more than 80 terms, however.) These terms appear to be roughly sorted in decreasing order of relevance or generality.



1-10 of 1,497,480 hits for **renewable energy**


Refine your search

- energy technologies
- renewable energy sources
- renewable energy technologies
- wind energy
- wind power
- renewable sources
- geothermal
- energy resources
- sustainable energy
- wind turbines
- solar power
- turbines
- renewable energy systems
- renewable energy resources
- energy systems
- biomass energy
- renewable energies
- non-renewable

Figure 4. Example of "Refine your search" results provided by Scirus

We can extract these terms in an automated way using our software, but 80 terms is far short of our desired number of terms, however. To use Scirus to collect more terms, we go through several steps as follows.

1. We perform follow-up searches on some or all of the returned terms, and extract the terms related to those follow-up terms.
2. This process forms a tree of terms with "renewable energy" at the root, its related terms at the second level, and so on. If all 80 returned terms were expanded, the tree would grow in width very quickly before getting very deep.
3. Since our goal is to discover terms that are highly specific and not necessarily well-known, we would like our term tree to grow deep more so than it grows wide. The depth/width tradeoff is left as an adjustable parameter in our software. We specify this parameter as a list whose length is the number of levels in the tree, and whose elements represent the branching factor at each particular level of the tree. So for example, if our list were [20, 10, 5], we would expand the 20 most relevant terms at the first level in our search, the 10 most relevant terms to each term in the second level of our search, and the 5 most relevant terms in the third level of our search. We are currently using the pattern [15, 5], which, after removing duplicates, returns approximately 3,000 terms. We have used this pattern because it seems to provide a good balance between relevance and specificity, while also returning approximately the number of terms we are looking for. We have also experimented with three-level patterns, but have

found that terms at the third level are often irrelevant. Section 4.1 of this report describes a proposed solution to this issue.

4. Note: Even when a term is not expanded for a follow-up search, it is still added to our list of terms that we are collecting. So in our example, each of the 15 follow-up terms at the first level returns not 5, but 80 terms of its own. 5 is the number of terms that get further expanded to return 80 of their own.

Finally, we merge the terms collected from Compendex, Inspec, and Scirus into one list. After removing duplicates, our 1,600 Compendex terms, 1,400 Inspec terms, and 3,000 Scirus terms become approximately 5,600 terms in total. It is notable that the three databases give us only approximately 400 duplicates, compared to 5,600 unique terms. It is due to this uniqueness of the three databases that we feel compelled to collect and combine terms from all of them. An open area of investigation involves deciding on a good number of terms to collect from each database. We need to manage the tradeoff between our desire to collect as many useful terms as possible, and our desire to contaminate the term list as little as possible with irrelevant terms.

2.2 Step 2 - Determination of hit count trends

In this step, we determine the prevalence trend over time of each of the terms collected in the previous step. Specifically, we use the hit count – the number of articles identified from a publication database when queried with the term – to represent prevalence. To determine the hit count trend over time, we restrict our search to articles published in a given year. We perform a search for each year we are interested in, so that we have a record of how the number of articles published about the given topic has changed over time.

2.2.1 Maintaining relevance

In determining the hit count for a particular term, we can try to insure that the term is taken in its proper context by actually determining the hit count for [term] + ‘renewable energy.’ As an example, the term ‘corn’ is relevant to energy technologies for its role in ethanol production. A Scirus search for ‘corn’ returns approximately 2.1 million hits. A search for ‘corn’ + ‘renewable energy,’ on the other hand, returns only about 90,000 hits. The discrepancy is largely due to the vast number of articles about corn that have nothing to do with energy. For our purposes, 90,000 is the more accurate count. There are two potential downsides to requiring that ‘renewable energy’ be included in our queries, however. One issue is that there may be terms that are important in the energy field, but are rarely mentioned in articles that also mention the term ‘renewable energy’ (or even ‘energy’ for that matter). Examples include technologies that have a wide range of applications, one of which is energy-related. ‘Genetic engineering,’ for example, is an important component of biofuel development, and we may be interested in articles that describe the development of genetic engineering without any mention of the energy application. The second issue is that in some cases, a term appears in renewable energy-related articles that do not actually mention the term ‘renewable energy’ because it is clear from the context. An article about windmills, for example, is clearly relevant to renewable energy even if it never explicitly uses that expression. So an open question being investigated is whether or not to require that ‘renewable energy’ be included in our hit count queries.

2.2.2 Use of multiple sources

Just as we can use one or more sources to produce the list of potentially interesting keywords, we can use one or more database in determining the hit count trends of a topic. It is also important to note that the source(s) that we use for trend analysis need not be the same as those used to produce the list terms in the previous step. These are two independent processes; we can collect terms with whatever sources we choose, and we can analyze their prevalence trends using whatever sources we choose.

In deciding what databases to use for hit count trends, we need to consider the size and focus of the database, as well as the feasibility of using it for automated hit count collection. In general, the larger the database's collection of articles on renewable energy, the more statistically significant hit count trends will be. As far as feasibility, the database must allow us to perform thousands of queries with a high frequency, and must display hit counts in a manner that is extractable with software. With these considerations in mind, we are currently using Compendex and Scirus to gather hit count trends, as they tend to give relatively large hit counts even for obscure energy-related terms. In contrast, we have decided that Inspec, the third database we use in the term collection step, is too small of a database to be effective in this hit count collection step.

With both Compendex and Scirus, collecting hit counts for each term is a slow process. For example, finding hit counts for 5,000 terms over three years, which basically requires 15,000 queries to the data bases, takes about 5 hours to complete. However, these numbers tend to remain relatively unchanged from trial to trial, so we can store the results from one trial in a local database for instantaneous lookup later. This can prove useful if, for example, we choose to modify our trend analysis methods (section 2.3 of this report), and do not wish to spend time re-gathering terms and hit counts prior to the analysis.

Since we are looking for modern, upcoming technologies, we have focused on the years 2006, 2007, and 2008. In future work, we might extend these dates further into the past to get a longer picture of the trends of these technologies.

2.3 Step 3 - Identification of potentially interesting terms by rate of growth

Using the hit count trends we recorded in the previous step, we now want to choose from that large initial list a subset of terms that seem to fit the early growth description. There are several possible ways to characterize an early growth technology in terms of its hit count trend. We choose to rank each term according to the ratio of its final hit count to its initial hit count. The approximately 100 top-ranked terms will then be used in the final step of our methodology, which performs more in-depth analysis on those terms. With this ranking method, we choose to ignore intermediate (2007) hit counts. In future developments, we may use that information in our trend analysis as well. The use of intermediate years' hit counts may be particularly important if we use a longer timeline than 2006-2008.

By ranking terms according to the ratio of final hit count to initial hit count, we are strongly favoring terms with initially low hit counts. For example, a term whose hit counts increased from 1 to 3 will be ranked higher than a term whose hit counts increased from 100,000 to 200,000 over the same time period. This is in line with our interest in identifying technologies that are not yet well-known, but are on the steepest part of their growth curve.

An important issue that can emerge when finding the ratio of a term's final to initial hit count is that sometimes the initial hit count may be zero. There are a few possible workarounds to this problem. One possibility is to replace the zero with the lowest nonzero hit count the term acquires in a future year. If the hit count remains zero until the final year, then we accept that its ratio is infinite, and rank it against other infinite-ratio terms based solely on the final year's hit count. A second possibility is to define zero as some small number, say 0.1. The choice of this number is unfortunately somewhat arbitrary, but the smaller it is defined to be, the higher the ranking zero-hit count terms will receive. We are currently using the second method, but exploring various options is an important topic of upcoming work.

Another method we have experimented with is to first calculate the average hit count across all terms for 2006, and similarly for 2008. Then we look for terms whose 2006 hit count is much smaller than the 2006 average and whose 2008 hit count is much higher than the 2008 average. We have called this method "total growth" analysis (as opposed to "early growth.") While this method is interesting and potentially nformative, we believe that our early growth method is more practically useful. Yet another seemingly viable possibility for identifying interesting terms would be to rank them according to the slope of their growth, i.e. the ratio of the change in hit counts to the change in time. But since all terms have the same time range, this reduces to ranking them based on change in hit counts alone and this measure is not in line with our interests because ranking based on the difference in hit counts, as opposed to the ratio, favors terms whose hit counts were large to begin with.

There may be other metrics besides the ratio of the final hit count to the initial hit count that could suit our needs. For example, it might be a good idea to include intermediate years in the calculation as well. Defining precisely what is meant by "early growth" is an important part of this ongoing research.

2.4 Implementation of Initial Software

Our initial software is written in Python. It contains three modules, with each corresponding to one of the three steps of our methodology, i.e. term collection, hit count collection, and trend analysis.

1. The term collection module takes three parameters as input: a list of seed terms (which is often just a single term), the number of articles from which to collect terms in Compendex and Inspec, and the breadth/depth pattern to use for hierarchically collecting terms from Scirus. The software outputs the collected terms to a text file.

2. The hit count collection module takes as input the term file generated in the previous step, as well as a list of publication databases and the range of years from which to collect hit counts. The output of this module is a local database which contains, for each term, the hit counts for each year and for each publication database used.

3. The final module does the hit count trend analysis to identify our early-growth terms. It takes as input the local database produced at the previous step. For each publication database used to collect hit counts, this module performs the trend analysis and outputs a list of the best early-growth candidate terms, sorted by decreasing ratio of final hit count to initial hit count.

If we wish to collect around 5,000 terms, the first module takes approximately five minutes to complete this task. In the second step, if for each term we wish to collect three years' worth of hit counts from two different publication databases, this takes around five hours. The reason for the large increase over the time needed to complete the first step is that for each term we must visit and scan six html pages to collect that term's hit counts. We visit far fewer pages in the term collection step since each individual page provides many terms. The third module does not involve any reading of web content, so it completes nearly instantly. The bottleneck in our software is therefore the second step. We have performed some optimizations to make it more efficient, such as scanning only the section of a search results page that we know to contain the hit count, but this step is inherently time consuming.

3. Preliminary Results

Preliminary results for these first three steps are shown here.

1. As described above, we used Compendex, Inspec, and Scirus to gather approximately 5,600 terms on renewable energy**.**

2. Next, we used both Compendex and Scirus to identify hit count trends on each of the terms, appending each query with ‘renewable energy’ to keep the hit counts in the proper context.
3. Finally, for both our Compendex hit counts and Scirus hit counts, we ranked the terms according to the ratio of their 2008 hit count to their 2006 hit count.

The top 20 terms for both Compendex growth and Scirus growth are shown below.

Top Compendex growth:
energy, engineering, technology, wind, fuel, time, development, efficiency, cells, experimental, policy, oil, industry, economic, concentration (process), electric power systems, speed, economics, emission, wind energy

Top Scirus growth:
biopower, concentrating solar power (csp), genome sequence, biochemical pathways, equitable distribution, sustainable resources, homer, food prices, wind power prediction, plant sciences, small hydroelectric plants, sustainable communities, assortment, wind load, fischer-tropsch diesel, graph theory, mesoscale model, electric loads, european union energy policy, investment strategy

The results show some promising terms and some questionable ones. For example, ‘biopower,’ ‘concentrating solar power (csp),’ ‘biochemical pathways,’ and ‘fischer-tropsch diesel’ may be the kind of early growth technologies we are looking for. However, terms such as ‘energy,’ ‘engineering,’ and ‘technology’ are far too general, and terms such as ‘time’ and ‘homer’ are largely irrelevant.

We will continue to revise our term collection process, as well as our term ranking process, to try and increase the proportion of quality terms identified with our methods.

4. Improvements to original Methodology

The methodology described above provided some interesting results and illustrated this approach was promising, but it also highlighted some problem areas. This led to several improvements to our original methodology, as described in this section.

4.1 Better Filtering Scirus terms for relevance

As described in section 2.1.2 of this paper, a Scirus search for a given seed term will only return at most 80 related terms. This requires us to perform follow-up searches on the returned terms, and proceed in a hierarchical manner. (In contrast, when we use Compendex or Inspec to gather terms, a search for a single seed term can provide as many related terms as we would like, so no follow-up searches are needed.) A downside to using Scirus to collect terms in this hierarchical manner is that as the depth of the term tree grows, the terms we find can quickly become very general and irrelevant to our goal. For example, if we search for ‘renewable energy,’ three of the 80 terms that come back are ‘stand-alone,’ ‘harnessing,’ and ‘national security.’ If we deepen our tree by extending these terms, grossly irrelevant terms are returned. A sampling of the terms returned from ‘stand-alone’ includes ‘command line,’ ‘adjustable,’ ‘decision-making,’ ‘high availability,’ and ‘default value.’ The results from ‘harnessing’ include ‘humans,’ ‘animals,’ ‘publishing group,’ and ‘right arrow.’ And the results from ‘national security’ include ‘freedom of information,’ ‘law school,’ ‘facial recognition,’ and ‘donate.’

Even if we extend some of the more relevant terms returned from a ‘renewable energy’ search, we still find several irrelevant terms at the next level in the hierarchy. For example, ‘energy systems’ is a good, relevant term returned from ‘renewable energy.’ But if we further extend this, we get results that include ‘sealed,’ ‘fiscal year,’ ‘non-stop,’ and ‘proton.’ And clearly, if we were to extend these terms to the next level after that, we would be out of the realm of renewable energy entirely.

So we have somewhat of a conflict. We want to gather terms that are specific and not immediately obvious given the seed term ‘renewable energy.’ But if we look only two or three levels deep into the Scirus term hierarchy, we find that the majority of terms are entirely irrelevant to our original topic. We propose to combat this issue by indeed collecting terms deep in the hierarchy, but then screening them for relevance. We can test for relevance by seeing if the term in question points *back* to our seed term or to some term close to it in the hierarchy. In other words, for a term in question, we can look at the 80 terms returned from Scirus when searching for that term. If one of those returned terms is ‘renewable energy’ (or possibly some term that itself points to ‘renewable energy’), then we can deem the term relevant. Further, if the term were found several levels *down* from our seed term, but then points *back up* to it, we can presume that it is both relevant and specific, which is exactly the class of term that we are looking for.

As an example, if we use ‘sustainable energy’ as our seed term, ‘fuel cells’ is among the terms returned at the first level of the hierarchy. Then, when we perform a follow-up search on ‘fuel cells,’ some terms such as ‘laptops’ and ‘proton’ are returned, but we also find some useful terms such as ‘hydrogen fuel.’ ‘Hydrogen fuel’ can be identified as such because Scirus lists ‘sustainable energy’ as one of the terms that it in turn points to. In contrast, ‘laptops’ and ‘proton’ do not point back to ‘sustainable energy.’

Figure 5 illustrates this process. It shows a representation of the Scirus term hierarchy with ‘Sustainable energy’ at the root. ‘Hydrogen fuel’ links back up the hierarchy to ‘Sustainable energy,’ so we declare it to be relevant. Note that in reality this network is highly interconnected; this simple representation ignores many links.

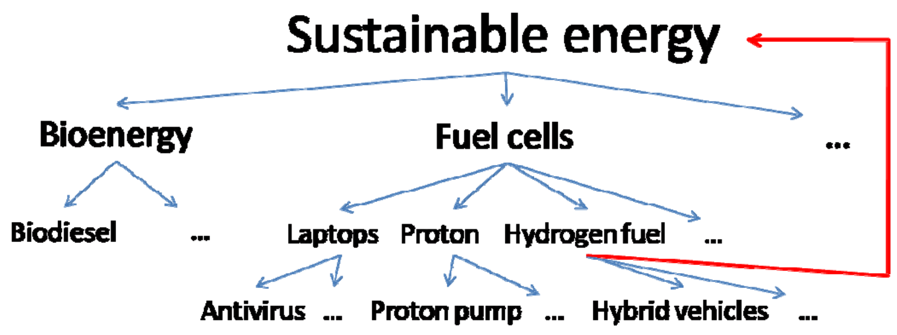


Figure 5. Representation of Scirus term hierarchy with ‘Sustainable energy’ at the root.

We present results here derived from this method of relevance filtering. We collected terms that were found in the first three levels of the term hierarchy from the seed term ‘renewable energy’, and that also pointed back up to ‘renewable energy.’ In other words, for each term we collected, ‘renewable energy’ was listed among the 80 terms Scirus listed as relevant to that term. We then gathered hit counts over time for each of these terms, and used our trend analysis to identify the top early growth candidates. The top 20 terms are listed here, in ranked order according to growth of Scirus hit counts:

pakistan, solar water, wind turbines, turkey, renewable energies, hydropower, stromerzeugung, mppt, solar thermal, solar water heating systems, photovoltaic systems, linear test bed, biodiesel production, ocean wave power, renewable, energy technologies, turbines, hydroelectric energy, electrification, megawatts

It can be seen that these terms are more relevant to the topic of renewable energy than are the terms found in our previous results. We can make further improvements if we relax the requirement that any term in our list points directly back to 'renewable energy.' In the results presented below, we instead allow any term that points back to either 'renewable energy,' 'sustainable energy,' or 'alternative energy.' The top 20 terms in Scirus growth that follow this modified requirement are:

electric cars, hydrogen vehicles, solar house, pakistan, gas engine, solar water, climate policy, solar energy systems, wind energy, tidal power, offshore drilling, renewables, solar energy applications, turbine noise, global warming, solar power, solar electricity, nanotechnology, alternative fuels, alternative energy

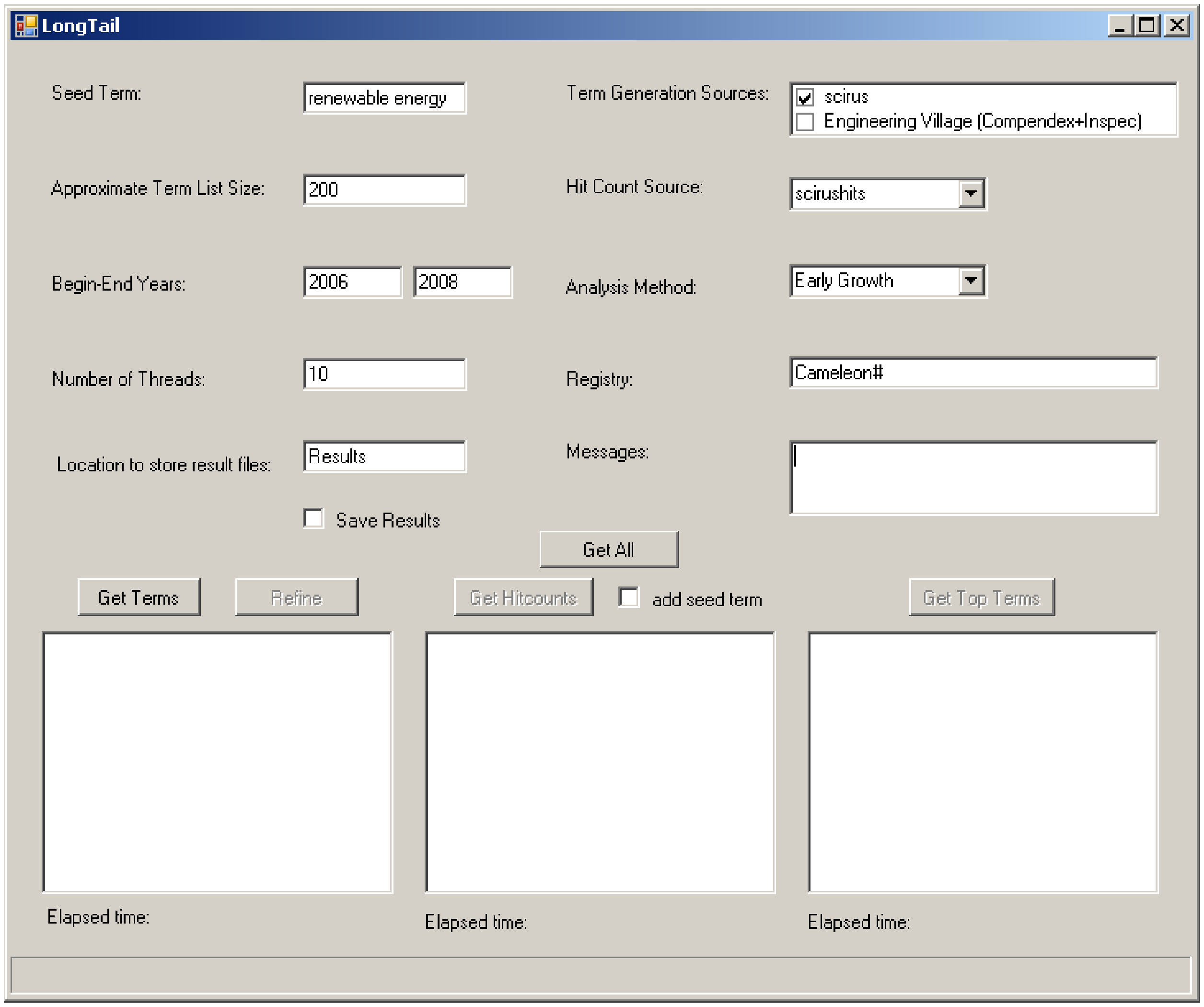
In this list, we see some very positive results. Many of the terms listed, including the top-ranked 'electric cars' and 'hydrogen vehicles,' describe energy-related technologies that are growing in popularity. In addition, we see some terms which describe energy issues rather than technologies, such as 'climate policy,' 'offshore drilling,' and 'global warming.' And while terms such as these are not precisely what we are looking for, they do represent current events which are becoming more important issues over time. The biggest problem with this list of terms is that they tend to be too general. For example, while electric cars are indeed quickly growing in popularity, we would like this list to contain specific technologies behind electric car design.

The approach described above is part of an active area of our research. Filtering for relevance could be made more robust with a more complex approach. For example, as we have seen, it may not be ideal to require a term to point directly back to the single seed term in order to be considered relevant. Instead, it may be enough to point to a term that is close to our seed, or to another term that has already been declared relevant. We are exploring different approaches, keeping in mind our goal of increasing both the specificity and the relevance of our resultant terms.

5. Early Growth Technology Analysis Tool

As was described above, we have developed a software program for identifying early growth emerging technologies by going through the steps of: (1) collect a large number of terms related to the field we are interested in, (2) record the change over time of hit counts from online publication databases for each of our collected terms, (3) use these hit count trends to reduce the size of our list to a shorter list of potentially interesting technologies. Although effective to producing the results described above, this software was not designed to be especially "user friendly." As a result, we have created the Early Growth Technology Analysis Tool with desktop and web versions to perform those steps automatically. Here we will briefly explain it.

Both the desktop and web versions of the tool have easy-to-use interfaces, as shown in Figure 4.


LongTail

Seed Term: renewable energy

Term Generation Sources: [✓] scirus [ ] Engineering Village (Compendex+Inspec)

Approximate Term List Size: 200

Hit Count Source: scirushits

Begin-End Years: 2006 2008

Analysis Method: Early Growth

Number of Threads: 10

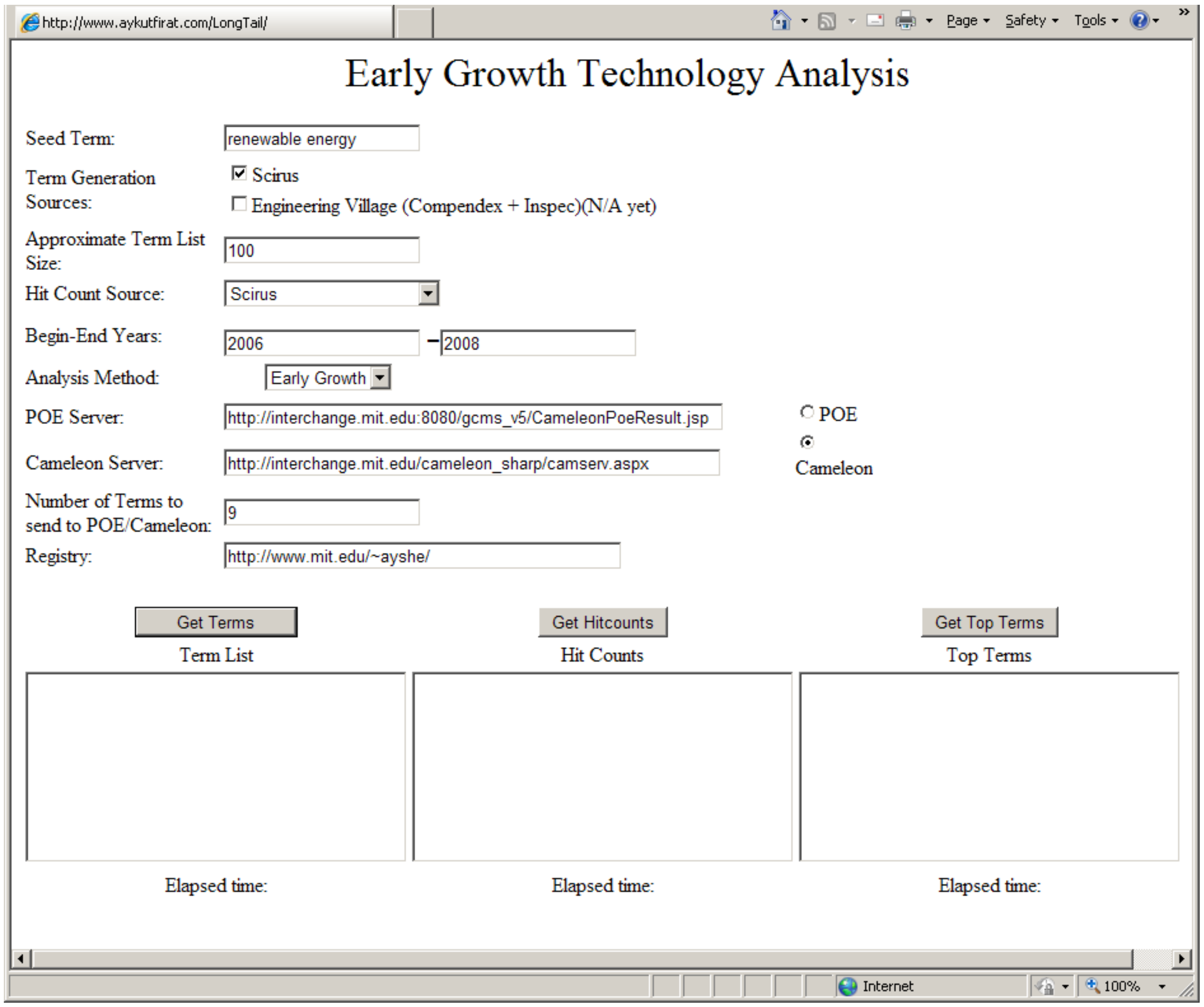
Registry: Cameleon#

Location to store result files: Results

Messages:

[ ] Save Results

4 a) Desktop Version of Early Growth Analysis Tool


http://www.aykutfirat.com/LongTail/

Early Growth Technology Analysis

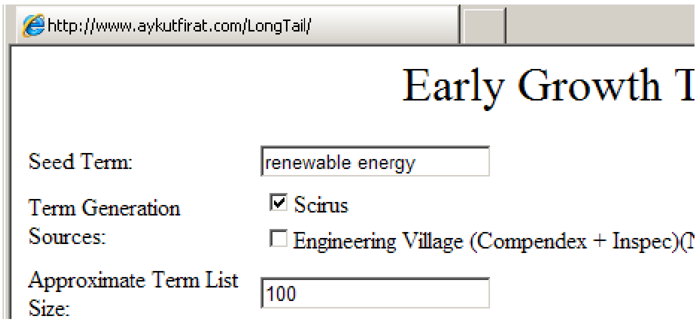
Seed Term:	renewable energy
Term Generation Sources:	☑ Scirus ☐ Engineering Village (Compendex + Inspec)(N/A yet)
Approximate Term List Size:	100
Hit Count Source:	Scirus
Begin-End Years:	2006 – 2008
Analysis Method:	Early Growth
POE Server:	http://interchange.mit.edu:8080/gcms_v5/CameleonPoeResult.jsp
Cameleon Server:	http://interchange.mit.edu/cameleon_sharp/camserv.aspx
Number of Terms to send to POE/Cameleon:	9
Registry:	http://www.mit.edu/~ayshe/

○ POE [✓ ● Cameleon]

4 b)Web Application Version of Early Growth Analysis Tool

Figure 7. User interface to Early Growth Analysis Tool

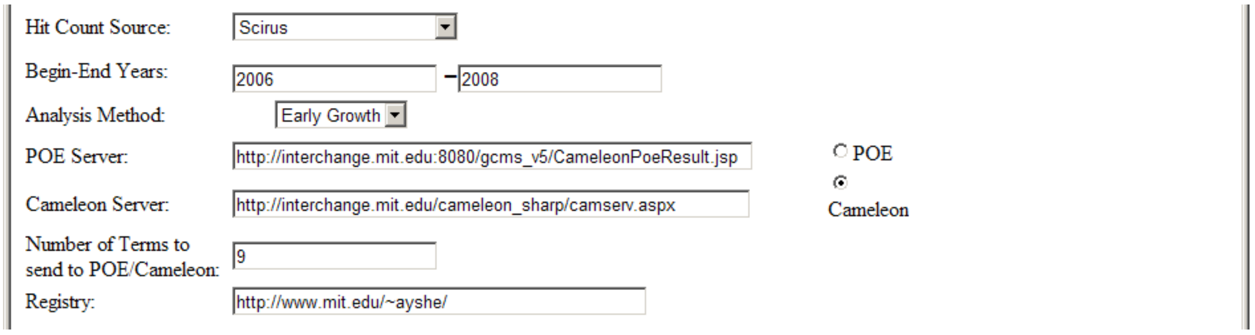
The first row in the tool is for the Seed Term where the user enters the field of interest, such as renewable energy, biotechnology or geothermal. For now, Scirus and Engineering Village (Compendex + Inspec) are the available source choices for term generation. Next, the user enters the number of related terms desired to be gathered by the tool in the "approximate term list size" textbox. At this point all the data required for the first step (collecting terms) has been entered.


http://www.aykutfirat.com/LongTail/

Early Growth T

Seed Term:	renewable energy
Term Generation Sources:	☑ Scirus ☐ Engineering Village (Compendex + Inspec)(
Approximate Term List Size:	100

For the second step (collecting hit counts), the user selects a single hit count source, the beginning and the ending years, and either the POE or Cameleon Servers that will be used to automatically extract the hit counts. The required spec files that are shown below are taken from the registry. We use a pull down menu because only one hit source is allowed at a time.



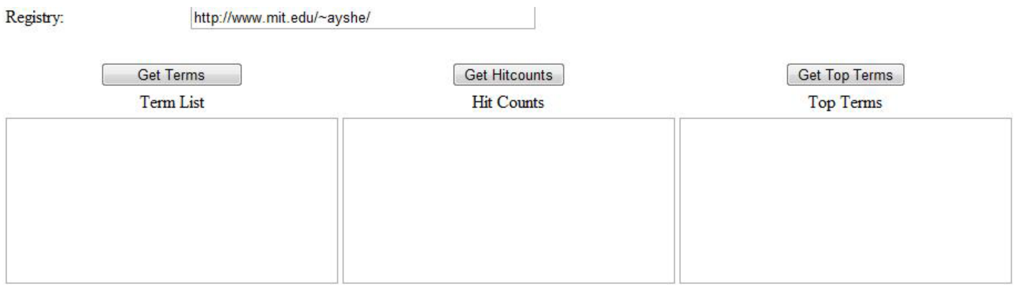
Hit Count Source:	Scirus
Begin-End Years:	2006 – 2008
Analysis Method:	Early Growth
POE Server:	http://interchange.mit.edu:8080/gcms_v5/CameleonPoeResult.jsp
Cameleon Server:	http://interchange.mit.edu/cameleon_sharp/camserv.aspx
Number of Terms to send to POE/Cameleon:	9
Registry:	http://www.mit.edu/~ayshe/

POE [✓ Cameleon]

With Cameleon Server we use threads to collect terms. 'Number of terms to send to POE/Cameleon' parameter defines how many queries are sent at once. With POE server we send a UNION query with the selected number of sub-queries and POE handles the rest. The Spec files used are shown in Appendix 2.

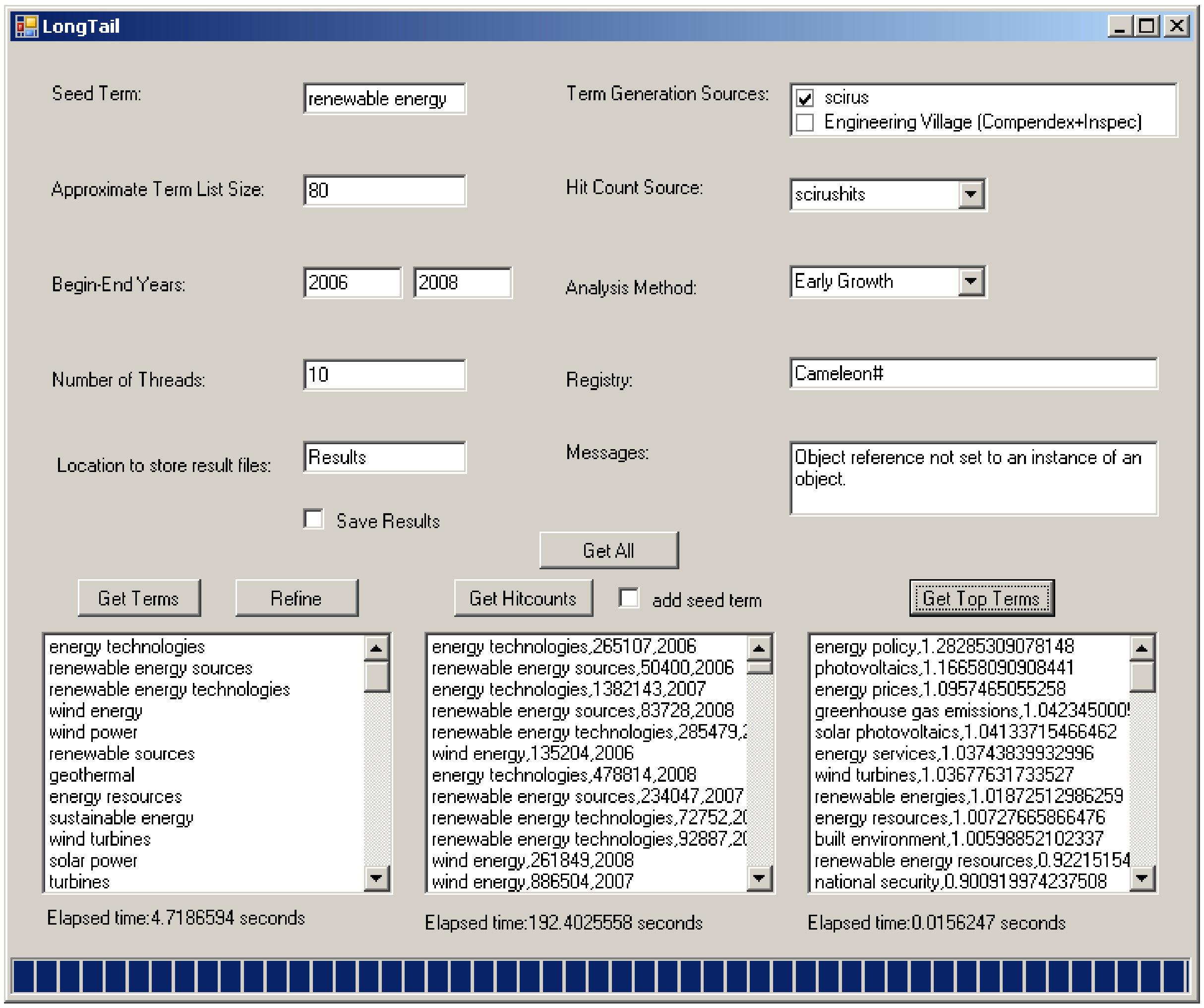
Finally, the user selects the analysis method for the third step: In Early Growth parameter, we rank each term according to the logarithm of the ratio of its final hit count to its initial hit count where we are favoring terms with initially low hit counts.

The user, now, can execute each step one by one, by clicking Get Terms, Get Hitcounts and Get Top Terms buttons. The elapsed time will be shown after each step under the related listbox.



Registry: http://www.mit.edu/~ayshe/

Below is a screenshot for an early growth analysis where the seed term is renewable energy:



Seed Term: renewable energy

Term Generation Sources: [✓ scirus] Engineering Village (Compendex+Inspec)

Approximate Term List Size: 80

Hit Count Source: scirushits

Begin-End Years: 2006 2008

Analysis Method: Early Growth

Number of Threads: 10

Registry: Cameleon#

Location to store result files: Results

Messages: Object reference not set to an instance of an object.

Save Results ☐

add seed term ☐

```
energy technologies
renewable energy sources
renewable energy technologies
wind energy
wind power
renewable sources
geothermal
energy resources
sustainable energy
wind turbines
solar power
turbines
```

Elapsed time:4.7186594 seconds

```
energy technologies,265107,2006
renewable energy sources,50400,2006
energy technologies,1382143,2007
renewable energy sources,83728,2008
renewable energy technologies,285479,2
wind energy,135204,2006
energy technologies,478814,2008
renewable energy sources,234047,2007
renewable energy technologies,72752,2
renewable energy technologies,92887,2
wind energy,261849,2008
wind energy,886504,2007
```

Elapsed time:192.4025558 seconds

```
energy policy,1.28285309078148
photovoltaics,1.16658090908441
energy prices,1.0957465055258
greenhouse gas emissions,1.042345000
solar photovoltaics,1.04133715466462
energy services,1.03743839932996
wind turbines,1.03677631733527
renewable energies,1.01872512986259
energy resources,1.00727665866476
built environment,1.00598852102337
renewable energy resources,0.92215154
national security,0.900919974237508
```

Elapsed time:0.0156247 seconds

REFERENCES

www.scirus.com

http://www.engineeringvillage2.org/controller/servlet/Controller

Aykut Firat, Stuart Madnick, Nor Adnan Yahaya, Choo Wai Kuan, and Stéphane Bressan, “Information Aggregation using the Caméléon# Web Wrapper,” Proceedings of the 6th International Conference on Electronic Commerce and Web Technologies (EC-Web 2005),” Copenhagen, Denmark, August 23 - August 26, 2005, also published in Springer Lecture Notes in Computer Science (LNCS) 3590, K. Bauknecht et al (Eds.), pp.76-86, Springer-Verlag Berlin, 2005 [SWP # 4562-05, CISL 2005-06],

Appendix 1: Information about Data Source

Early growth technology utilizes online publication database systems as information sources. Hit counts of technology terms collected from these databases are critical in our analysis. Therefore, the feature of databases will affect our analysis in many ways, for example, where are the resources come from, how do databases rank their results, and so on.

In this Appendix, we describe some of the information sources that we have studied: SCIRUS, Inspec, Compendex, Google Scholar, Web of Science and ScienceDirect. Among these, we mainly use Compendex, Inspec, Scirus, and Google Scholar for our analysis. Comparison of databases below shows their specific feature, thus gives us an idea of strength and shortage of each database and why we choose the databases we are using.

The coverage of publication databases is one of the important factor affecting our choice, only databases with appropriate size of collection can facilitate our analysis, so firstly, we discuss the quantity of coverage, which is the hit counts we get from each database. We searches on different databases for technologies we are interested. By trying several technology terms, we can get an impression of how large the size of the collection of publication resources of each database, which we call, the quantity of coverage.

Another important thing would be the scope of coverage, which means from where they collect these resources. Some databases may be highly academic focused, for they only collect thesis, papers in publication journals, or books. Others are more general, for they may contain related resources from any online information they can access, for example, academic papers, newspapers, and nonacademic websites.

The information, or rather, the pattern of searching provided by databases is the third thing we are interested in. For early growth technology analysis, a critical thing we aim to focus on is the time span. How early the search engine could extract resources from is what we are looking for. Other main information we also care about is whether they have abstract, available full text, citation list, and so on. Fourthly, we are interested in the elaboration of keywords. Some publication databases may have a fixed list of keywords from which authors can choose from, some may not. These keywords limitation and standard also affect our analysis in some degree.

Last but not least, we aim to find out the unique feature of certain databases. Every database has their special function or searching pattern which is possible for them to differ their searching result. Google Scholar is a good example. It differs most from all other databases in our list. Its searching pattern, unique information provided, coverage and other features makes it result different.

1. **Quantity of coverage**

The hit counts below are updated as of 03/04/2009.

Following is an example of comparison of searching the key word “renewable energy”, “greenhouse effect”, “artificial intelligence”, “solar cells” and “geothermal” in these sources. Numbers of results of our search are:

Renewable energy					
SCIRUS	Inspec	Compendex	Google Scholar	Web of Science	ScienceDirect
1,511,111	15478	27201	346,000	6,711	45,622
Greenhouse effect					
SCIRUS	Inspec	Compendex	Google Scholar	Web of Science	ScienceDirect
572,126	2801	10569	884,000	16,260	62,520
Artificial intelligence					
SCIRUS	Inspec	Compendex	Google Scholar	Web of Science	ScienceDirect
1,313,063	164230	176204	1,180,000	22,194	71,177
Solar cells					
SCIRUS	Inspec	Compendex	Google Scholar	Web of Science	ScienceDirect
734,567	43108	41562	707,000	21,389	83,317
Geothermal					
SCIRUS	Inspec	Compendex	Google Scholar	Web of Science	ScienceDirect
330,447	5757	16579	280,000	8,384	33,408

2. **Scope of coverage**

a. SCIRUS covers web sites, news, journals, web articles and academic papers with abstract and reference.

As provided on the webpage of SCIRUS, it is a science-specific search engine returning results from the whole Web. It currently covers over 485 million science-related Web pages, including:

- 156 million .edu sites
- 54 million .org sites
- 9 million .ac.uk sites
- 52 million .com sites
- 36 million .gov sites
- Over 143 million other relevant STM(scientific, technical and medical) and University sites from around the world

In addition to Web pages, Scirus indexes the following special sources (the numbers are approximate):

- 447,000 articles from American Physical Society
- 536,000 e-prints from ArXiv.org
- 42,000 full-text articles from BioMed Central
- 19,000 documents from Caltech Coda
- 3,300 e-prints from Cogprints
- 81,800 full-text articles from Crystallography Journals Online
- 24,000 documents from CURATOR
- 2.1 million documents from Digital Archives
- 24,000 documents from DiVa
- 98,500 full-text articles from Project Euclid
- 3,200 documents from HKUST Institutional Repository
- 56,000 documents from The University of Hong Kong
- 12,700 full-text documents available from IISc
- 11,000 full-text documents available from Humboldt Universität

- 284,000 full-text articles from Institute of Physics Publishing
- 23.1 million patent data from LexisNexis
- 16,000 full-text articles from Maney Publishing
- 40,000 full-text documents from MD Consult
- 585,000 full-text documents from Nature Publishing Group
- 18.1 million Medline citations via PubMed
- 72,000 documents from MIT OpenCourseWare
- 24,700 technical reports from NASA
- 792,000 full-text theses and dissertations via NDLTD
- 8,900 documents from Organic Eprints
- 1,690 documents from PsyDok
- 1.5 million articles from PubMed Central
- 738,000 documents from RePEc
- 63,000 full-text articles from Royal Society Publishing
- 619,000 full-text articles from SAGE Publications
- 8.0 million full-text articles from ScienceDirect
- 463,000 full-text journal articles from Scitation
- 9,100 articles from SIAM
- 16,600 documents from University of Toronto T-Space
- 21,800 full-text documents from WaY

As showed above, SCIRUS covers both general information from non-academic websites and specific research sources.

b. Inspec and Compendex mainly cover academic articles with abstract and reference.
As provided on the webpage of Inspec, it scientific literature in electrical engineering, electronics, physics, control engineering, information technology, communications, computers, computing, and manufacturing and production engineering. The database contains over eight million bibliographic records taken from 3,000 scientific and technical journals and 2,000 conference proceedings. Over 400,000 new records are added to the database annually.
Compendex contains over ten million references and abstracts taken from over 5,600 scholarly journals, trade magazines, conference proceedings and technical reports. Its coverage includes nuclear technology, bioengineering, transportation, chemical and process engineering, light and optical technology, agricultural engineering and food technology, computers and data processing, applied physics, electronics and communications, control, civil, mechanical, materials, petroleum, aerospace and automotive engineering as well as narrower subtopics within all these and other major engineering fields. Approximately 650,000 new records are added to the database annually from over 190 disciplines and major specialties within engineering. Compendex is updated weekly.

c. Google Scholar covers academic articles with abstract and reference, and also books.
As indicated on the webpage, Google Scholar provides a simple way to broadly search for scholarly literature. From one place, you can search across many disciplines and sources: peer-reviewed papers, theses, books, abstracts and articles, from academic publishers, professional societies, preprint repositories, universities and other scholarly organizations.

d. Web of Science mainly covers academic articles with abstract and reference.
As indicated on the webpage, Web of Science consists of seven databases containing information gathered from thousands of scholarly journals, books, book series, reports, conferences, and more.

e. **ScienceDirect** mainly covers academic articles with abstract and reference, or abstract and article outline.
As provided on the webpage, ScienceDirect contains over 25% of the world's science, technology and medicine full-text and bibliographic information in both books and journal formats.
Journals: Over 2,500 authoritative titles. The collection includes high-impact-factor titles such as THE LANCET, Cell and Tetrahedron.
Journal Backfiles **:** A historical archive of titles from 1994 and older, many going back to Volume1, Issue1. Together with more recent Backfiles there are more than nine million articles.
Online Books: The continuously expanding books collection on ScienceDirect includes eBooks, Major Reference Works, Book Series and Handbooks. Over 10,000 titles in total including backfiles.

3. Main information provided by each source

SCIRUS shows information about brief abstract, author, date, type of articles (whether they are common web page articles, academic papers or books), link to full text and link to similar results, which links below each record to view results that share your query terms in addition to the keywords pertaining to that particular result. The keywords of a record combine the author-assigned keywords with those automatically extracted by Scirus.

Inspec and Compendex have information of abstract, link to detailed full text, author, article source, dates, database linked to and link to MIT sfx (a link of local database collection of MIT) to get full text. As we can see, most are the same as SCIRUS, but the difference is that Inspec and Compendex contain only academic resources, so they do not need to show the type of articles, for it is apparently known. And also, we could get access to them from local (MIT) online library. For example, MIT sfx, as mentioned above, is an effective link to local database of MIT.

Google Scholar, however, has various types of information which depends on the format of resource (books, articles, general webpages, etc.) It has author, date, brief abstract, format (e.g. book, pdf, etc.), cited times, link to full text, link to related articles and link to key authors.
Web of Science has information of abstract, author, date, source, times cited, references, document type, language, author keywords, reprint address, addresses, E-mail addresses, publisher and links to MIT sfx for full text.

ScienceDirect has information of abstract, article outline (it is a catalogue of the article), author, date, source, link to related articles. Some records have links to full text in PDF format.

4. Elaboration of types of keyword provided: Author provided keywords or not.
Only Inspec, and Compendex provide both fixed list of keywords and author provided keywords, which they refer to as “controlled terms” and “uncontrolled terms”. Other databases do not have this feature.

5. Other information provided

SCIRUS can sort results either by relevance (how close is the text related to our search key word “renewable energy) or date. It can filter search results by content sources (e.g., restrict to journal sources, preferred web) or file types (PDF, Word, HTML). The search could be refined to similar key words (e.g. energy technologies, renewable energy sources, etc.) It also has a different link to news results. In the results, a large amount of academic papers found are linked to ScienceDirect.

Inspec and Compendex are imbedded in Engineer Village, therefore, they have similar features most of the time. Results of these two databases can be refined by including or excluding author, author affiliation, controlled vocabulary, classification code, country, document type, language, year, serial title and publisher. In Inspec, the year could go back to as far as 1937; while in Compendex, it is 1909. But the coverage of Compendex is broader than Inpsec. Compendex contains nuclear technology, bioengineering, transportation, chemical and process engineering, light and optical technology, agricultural engineering and food technology, computers and data processing, applied physics, electronics and communications, control, civil, mechanical, materials, petroleum, aerospace and automotive engineering as well as narrower subtopics. Inspec, however, contains electrical engineering, electronics, physics, control engineering, information technology, communications, computers, computing, and manufacturing and production engineering.

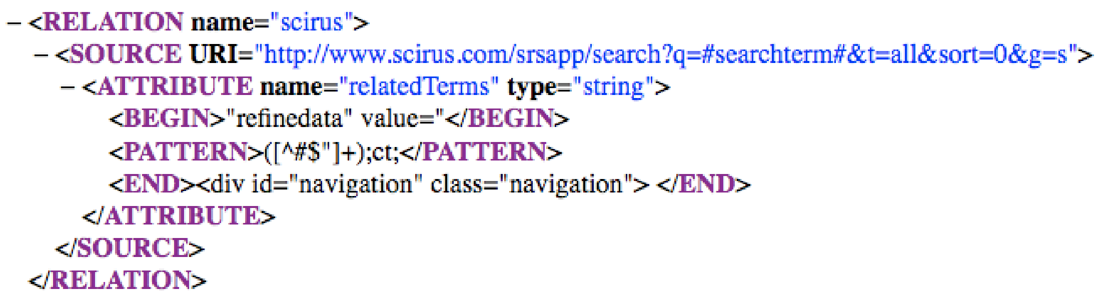
Web of Science provides information about number of times the searched article has been cited by, related records based on shared references and links to the journal (in which the searched article is published)'s impact factor. In addition, customers can also view the record in other databases.
In ScienceDirect, searched results can be refined to content type (journal, book or reference work), journal/book title and year. The year when searched records published can go back to 1989 and earlier.

Google Scholar differs very much from databases above. It has most variable types of resources. And the results link to different databases (ScienceDirect is most frequently linked to). Sometimes it also links to google book search, if the result we get is originally in the format of a published book. Google Scholar ranks its search results in the following way: weighing the full text of each article, the author, the publication in which the article appears, and how often the piece has been cited in other scholarly literature. The most relevant results will always appear on the first page.

Appendix 2: Specification (spec) files used to extract information from sources

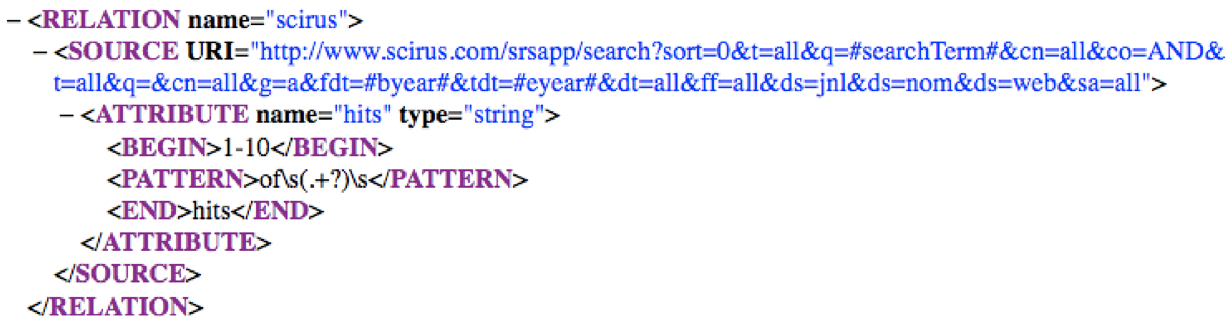
We use Cameleon # in the Early Growth Analysis Tool. Cameleon# is a generic web wrapper developed by the MIT Context Interchange (COIN) group. Web wrappers in the COIN group are used to treat semi-structured web data as ordinary relational data sources that can be processed using the standard SQL query language. The common element of all of the COIN wrapper development toolkits is that they separate the extraction knowledge from the code by expressing the former in a separate specification file (spec file). The code remains untouched and web sources are wrapped by creating a simple text file expressing extraction rules such as what URL(s) to visit, and what patterns to apply, and so on.

A sample spec file is shown below in XML format. In this spec file, the Web address of Scirus is indicated in the SOURCE tag. The input attributes (searchTerm, bYear, and eYear) are enclosed within # signs, and are expected from the user. The BEGIN and END tags specify (in regular expressions) landmarks preceding and following the data of interest. Finally the pattern specifies the regular expression for the data to be extracted.



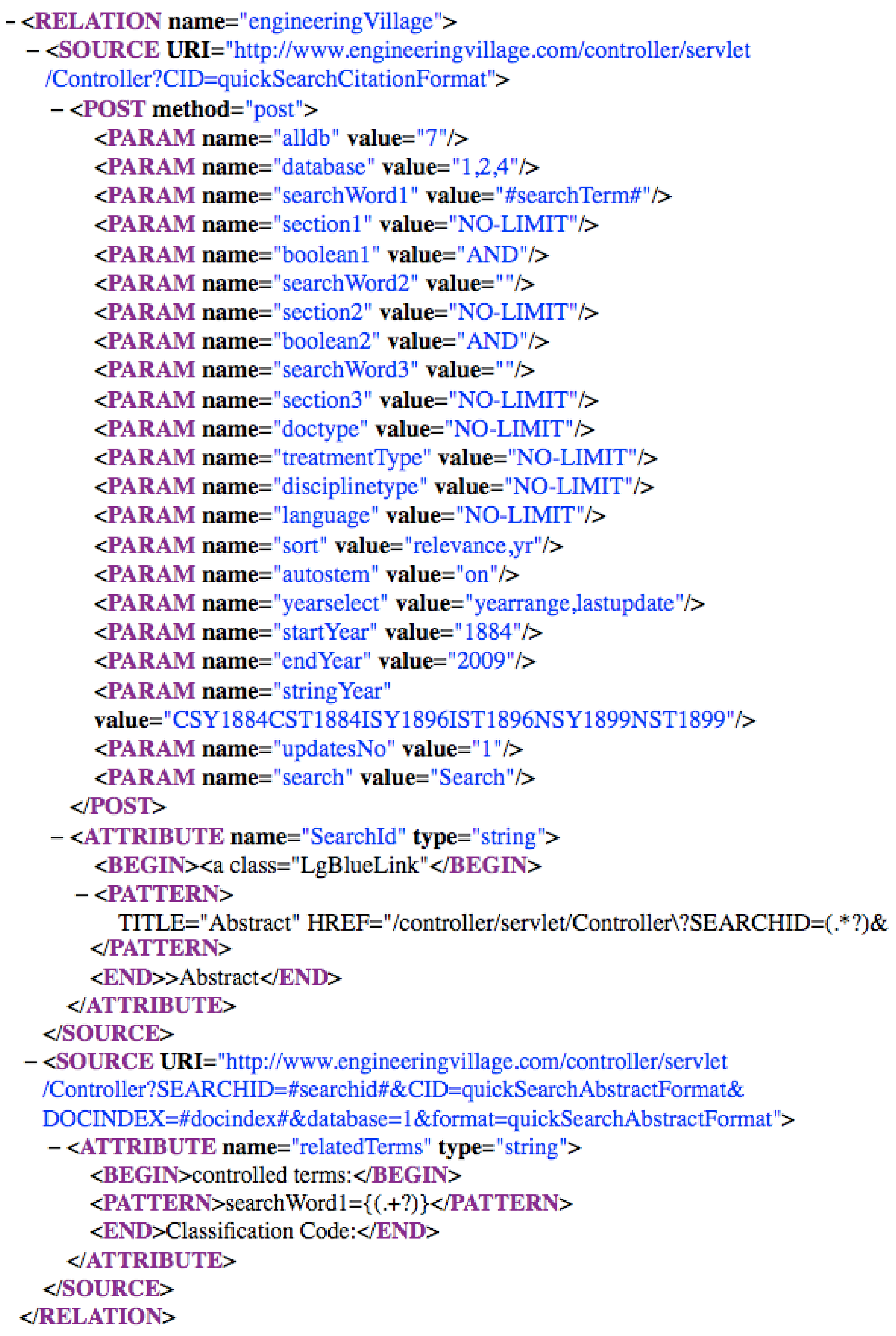
```
– <RELATION name="scirus">
  – <SOURCE URI="http://www.scirus.com/srsapp/search?q=#searchterm#&t=all&sort=0&g=s">
    – <ATTRIBUTE name="relatedTerms" type="string">
        <BEGIN>"refinedata" value="</BEGIN>
        <PATTERN>([^#$"]+);ct;</PATTERN>
        <END><div id="navigation" class="navigation"> </END>
      </ATTRIBUTE>
    </SOURCE>
  </RELATION>
```

scirus.xml



```
– <RELATION name="scirus">
  – <SOURCE URI="http://www.scirus.com/srsapp/search?sort=0&t=all&q=#searchTerm#&cn=all&co=AND&
    t=all&q=&cn=all&g=a&fdt=#byear#&tdt=#eyear#&dt=all&ff=all&ds=jnl&ds=nom&ds=web&sa=all">
    – <ATTRIBUTE name="hits" type="string">
        <BEGIN>1-10</BEGIN>
        <PATTERN>of\s(.+?)\s</PATTERN>
        <END>hits</END>
      </ATTRIBUTE>
    </SOURCE>
  </RELATION>
```

scirushits.xml



```
- <RELATION name="engineeringVillage">
  - <SOURCE URI="http://www.engineeringvillage.com/controller/servlet
    /Controller?CID=quickSearchCitationFormat">
    - <POST method="post">
        <PARAM name="alldb" value="7"/>
        <PARAM name="database" value="1,2,4"/>
        <PARAM name="searchWord1" value="#searchTerm#"/>
        <PARAM name="section1" value="NO-LIMIT"/>
        <PARAM name="boolean1" value="AND"/>
        <PARAM name="searchWord2" value=""/>
        <PARAM name="section2" value="NO-LIMIT"/>
        <PARAM name="boolean2" value="AND"/>
        <PARAM name="searchWord3" value=""/>
        <PARAM name="section3" value="NO-LIMIT"/>
        <PARAM name="doctype" value="NO-LIMIT"/>
        <PARAM name="treatmentType" value="NO-LIMIT"/>
        <PARAM name="disciplinetype" value="NO-LIMIT"/>
        <PARAM name="language" value="NO-LIMIT"/>
        <PARAM name="sort" value="relevance,yr"/>
        <PARAM name="autostem" value="on"/>
        <PARAM name="yearselect" value="yearrange,lastupdate"/>
        <PARAM name="startYear" value="1884"/>
        <PARAM name="endYear" value="2009"/>
        <PARAM name="stringYear"
        value="CSY1884CST1884ISY1896IST1896NSY1899NST1899"/>
        <PARAM name="updatesNo" value="1"/>
        <PARAM name="search" value="Search"/>
      </POST>
    - <ATTRIBUTE name="SearchId" type="string">
        <BEGIN><a class="LgBlueLink"</BEGIN>
      - <PATTERN>
          TITLE="Abstract" HREF="/controller/servlet/Controller\?SEARCHID=(.*?)&
        </PATTERN>
        <END>>Abstract</END>
      </ATTRIBUTE>
    </SOURCE>
  - <SOURCE URI="http://www.engineeringvillage.com/controller/servlet
    /Controller?SEARCHID=#searchid#&CID=quickSearchAbstractFormat&
    DOCINDEX=#docindex#&database=1&format=quickSearchAbstractFormat">
    - <ATTRIBUTE name="relatedTerms" type="string">
        <BEGIN>controlled terms:</BEGIN>
        <PATTERN>searchWord1={(.+?)}</PATTERN>
        <END>Classification Code:</END>
      </ATTRIBUTE>
    </SOURCE>
  </RELATION>
```

engineeringVillage.xml

SQL Queries Used

Sample Queries:

1.
```
select relatedTerms from scirus where searchTerm= “biotechnology”
UNION
select relatedTerms from engineeringvillage where searchTerm= “biotechnology”
```

2.
```
select hits from scirus where searchterm= “agricultural biotechnology” and  byear= “2006”and
eyear= “2007”
UNION
select hits from scirus where searchterm= “animal biotechnology” and  byear= “2006”and eyear=
“2007”
```