Advanced Search Tools for Online Resources
UROP (Spring 2005) Final Report
April 06, 2005

By: Justin Chase
Faculty Advisor: Dr. Ted Pedersen

This project had two main goals. One was to create an interface that made advanced Google searching easy and the other was to implement this utility on both windows and Linux using .NET and Mono. This project was mostly successful overall. The advanced searching interface was created successfully and worked properly and has many helpful features which I will get to soon but the Mono implementation proved to be very difficult; I will explain this more later.

There is a web site for this project (http://www.d.umn.edu/~chas0084/UROP/) where the most recent version of code and documentation can be located. One of the most interesting aspects of this project is that I broke up all of the main features into separate libraries. There are two libraries and an interface to drive the functions of those libraries. There is a separate interface for both Linux and Windows; Mono is not yet able to recreate Windows forms, therefore I had to create a separate interface for both platforms.

At the heart of this project is the Google API. Google offers a free web service that allows any developer who signs up for an API key to be able to access their search engine programmatically. My first, and most useful, library created for this project is a Google API wrapper library. It wraps up the API functions and allows a developer to more easily use the Google API in a multi-threaded environment. This library is setup to use events as a way of communicating with a parent thread. Additionally, there are very helpful classes designed to help developers enumerate all of the language, topic and country restrict codes. These codes can be very unintuitive and the list is fairly lengthy, this feature is very handy for users who intend to use alternate language and country search features regularly. The only limitation has to do with obtaining cached data from Google; I left out those features for now since I wasn't planning on using it. I would like to add this feature eventually though.

The second library has to do with the advanced search tools. This library requires the Google search wrapper library to use. You give it a query string and it will perform multiple Google queries and return an extended list of URL's as well as a list of 100 keywords. These keywords are either one or two word combinations that are extracted from the given URL's that seem to correlate most closely to what a user is searching for. The idea behind this is that these keywords might help give a user a better idea of how to find what they are looking for as well as suggest new keywords for their query.

The interface for the windows version seems to have come together quite nicely and most major bugs are worked out. The interface does feel a bit busy to me and there are probably better ways of laying out all the features but it's pretty good as is. The interface for the Linux version is quite different however. I attempted to implement Mono on Fedora Core 3 and had some serious troubles getting it installed. After finally getting it installed I found that developing interfaces was not only many times more difficult it was also impossibly buggy. The simple interface I developed would simply stop working randomly and was impossible to debug since it would never happen at the same place twice. In the end I succumbed to living with a command line application for Linux. Perhaps, in the future, a graphical interface will be possible but for now Mono does not support this feature very effectively. As disappointing as this is, I suppose it was partly the purpose of this project to discover how well it could be done; regardless of the outcome.

The search libraries I created are very helpful for any future projects that wish to implement Google searching as a feature of that software. Interestingly enough I have been hired by Dr. Rajiv Vaidyanathan, Associate Professor in the School of Business and Economics to work on a related piece of software. This piece of software will use the Google library I created to search the internet for stock market data in an attempt to see how effectively the internet might be able to reveal certain stock trends. It's very interesting to find a project that is so closely related to advance searching, this quickly after this project.

One other aspect to this project that I'm happy with is the nDoc (http://ndoc.sourceforge.net/) documentation produced for this project. It's a very effective way to document your projects and comes in handy when attempting to develop additional software making use of those libraries.

This project was definitely a learning process for me, the most amazing discovery for me was that Mono does not even need to recompile .NET libraries to run. It's pretty amazing how they got that to work but it's about the only thing amazing about mono. I guess I was very interested in the basic idea of Mono but in the end it seems to me like it's a bit too fragmented and it deviates from the basic and clean way that Microsoft organizes and implements .NET. Mono tries to implement all sorts of other things, such as Gtk, and it ends up being very confusing and hard to work with, not to mention buggy. I almost like the idea of using Glade to generate XML forms that can be used as an interface for any language… but it's really more hassle than its worth. That may mostly be due to the fact that the Gtk interface model is very limited and confusing compared to other interfaces I've developed in the past. Mono is a serious project in progress and it seems like they're not advancing as fast as they should be and their so called "stable" releases are very buggy still. Currently my development IDE doesn't compile at all and I have no idea why. I've had a heck of a time getting the installations to just work. I should be able to produce an excellent command line utility using mono however.

In regards to the Google searching it was indeed a very great discovery process. I learned how to make use of Web References in .NET and honestly that is an amazing thing. It's so incredibly easy to use and the perfect way to distribute a resource such as the Google API. This project was also quite an experience with multi-threading. There are quite a few threads going on in the advanced searching library and it is all fairly well synchronized. It takes quite a bit of time to do all the queries linearly and ends up working much more efficiently as separate threads.

Additionally I felt like the keyword generating worked very well! I was very pleased with the quality of results and find it very helpful. If you go down a given set of keywords you can find a relation for almost every word quite easily. The addition of bi-gram keywords was an interesting idea by Dr. Pedersen and it also works quite well. I tried a tri-gram keyword generator but it ended up taking much longer to process and would rarely yield any tri-gram results anyway. It seems the bi-gram and single keywords worked the best.

Overall I think this project turned out very well, I'm happy with the results. Though, this project is probably not exceptionally exiting for most people at first glace, from a technical point of view it is very interesting and I have been able to bring up some very promising results using these libraries. Additionally, some people have reported that they have learned some new searching techniques that they were unaware of before. Also the libraries used for this project are very useful individually and I plan to continue to modify them.