# Introduction

The research for knowledge and information in the past was always complicated. One always had to go to a library and search for the appropriate book the information needed was written into. So the process of gaining a piece of knowledge was in fact long and complicated. This is still the case while writing scientific papers which have to be properly quoted. Yet in the last decade, quick information retrieval has become much simpler.

Today, information at the fingertips is the slogan that describes the development of search engines on the internet. People can gain knowledge and information just by opening their browser and typing a query into a search engine.

Never the less, there are different kinds of search engines which retrieve different kinds of information in different output forms.

In order to get an overview over the most common types of information retrieval, this paper will deal with three different search engines: Wolfram Alpha, Google and Greplin. Each of these three engines has a unique point in their strategy which alters the type of knowledge one can gain. In order to get an overview, the paper will explain each engine and point out differences between them. After that the paper will conclude the results of the analysis of the three engines.

# Wolfram Alpha

Wolfram Alpha is an application for artificial intelligence in the web. It is a new approach on retrieving knowledge by using Algorithms to interpret human language and form it into computer interpretable queries for the databases.

In order to achieve this, a strong base of mathematical formulas and algorithms is required. This base of algorithms is the core behind the intelligence of Wolfram Alpha. This part of the paper will explain how this base was developed, what it roughly includes and who this is deployed in the application itself.

Wolfram Alpha documents itself as being published in 2009. The development of the application started in 2005, as the company approached the idea of deploying a further use of their program Mathematica. Until this day, Mathematica delivers the solid base of Wolfram Alphas algorithms in terms of mathematical formulas and their deployment in complicated logical algorithms.

Today, Wolfram Alpha is used by various applications in order to retrieve data. Examples are SIRI, an Apple application, the Android equivalent IRIS or the search engine “bing” all base partially on the output of Wolfram Alpha queries. So it could be said that Wolfram Alpha already produces valuable output for users.

There was some criticism issued in the discussion about Wolfram Alpha in the past two years. The biggest issue was a discussion about the smartphone app which was released in 2009. The initial costs were $50. The company also blogged the mobile formatting option for this site as the functionality was identic to the app functionality. So as an answer to the arising criticism, Wolfram Alpha lowered the price of the app to $2 and offered a refund to customers who already bought the app. Also the mobile formatting option was reactivated.

By looking at the web-based face of Wolfram Alpha today, one sees only a textbox, but there is so much more behind the scenes that make this textbox one of the major developments in the field of artificial intelligence in the past years.

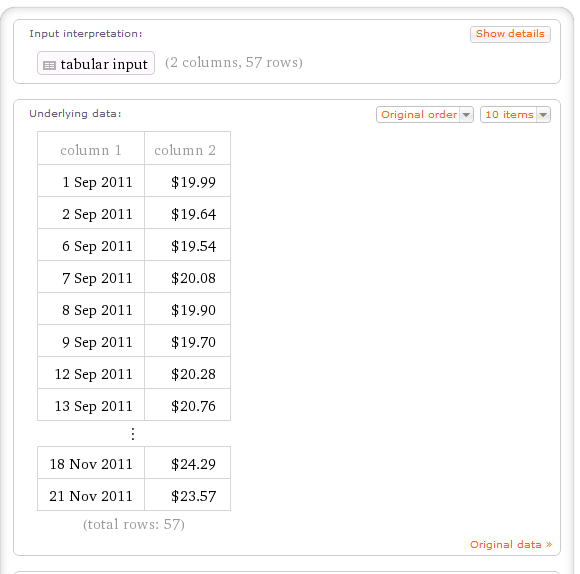
The first thing someone sees is that there can be multiple input types. As Wolfram Alpha states to be intelligent, it also promises to be able to process data in different forms. These forms are: normal text input, raw data input and images.

The differnt input types can be chosen by the following buttons. 

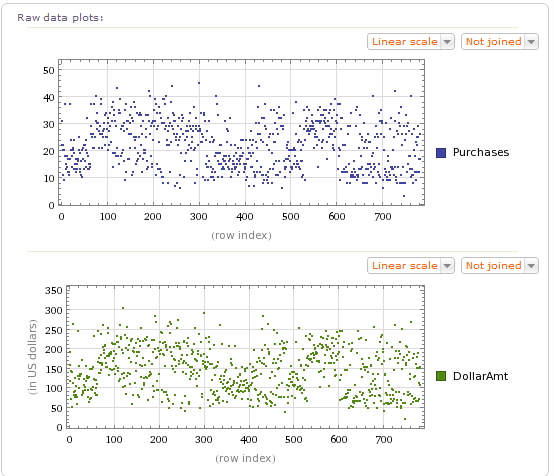
First off, this paper will discuss the raw data input. After that the image input will be looked at. The last task is to analyze the text input.

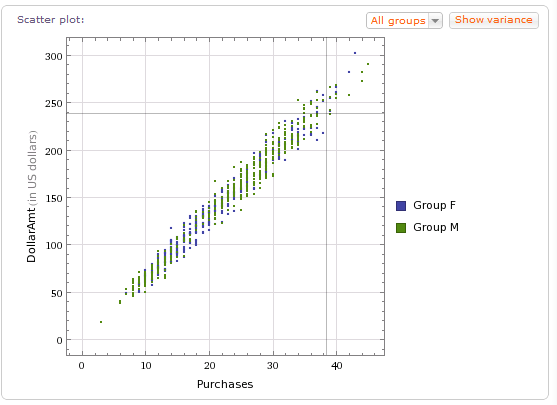
If one heard about the topic of data mining, the output of the input type of raw data will be familiar. It processes the data with mining algorithms in order to produce understandable graphics and ratios for the reader.

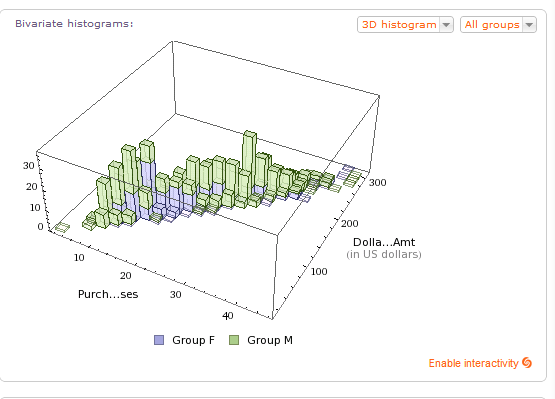
The outcomes of this raw data input are very extensive. Firstly, the data itself is displayed in a tabular format. The next information is Meta data of the table (e.g. number of rows, columns etc.).

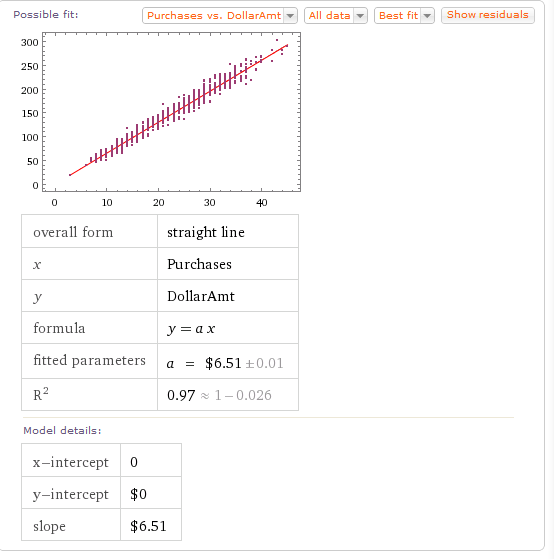


The next information is graphical approaches. This contains data plots, scatter plots, histograms in 2d and 3d and regression lines. Underneath are examples for those graphic types.



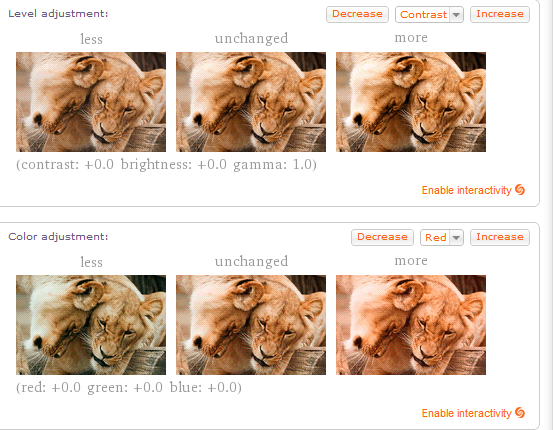






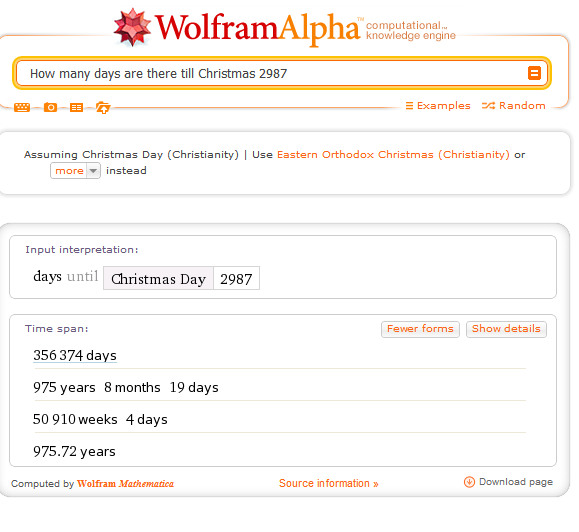
Besides the classical data mining approach, Wolfram Alpha also tries to link data to user-relevant data from its databases. This could be geographical data or data about music or persons (e.g. linking the data to a map). After this aspect, the engine computes various ratios to give the user information about mathematical concepts behind the data (e.g. regression formulas).

The next input which should be looked at is the image input. First of all, an image has to be uploaded to the Wolfram Alpha web side. After that the image can be linked in the provided textbox. After clicking the button to research it, the engine starts the analyses. The output of this process is not as far developed as the data and text input. But it is a new approach which is useful in some ways. It shows the segmentation of the picture, the dominant colors, the Meta information about the file and key points in the frame. It also provides some alternatives with different adjustments for the picture. This can be seen with different lighting, different color levels and different effects on the picture. With the Wolfram Alpha pro version, there are more ways of displaying the picture analysis available.



The last input to analyze is the most difficult one in development. The text input of Wolfram Alpha is one of the latest developments in the field of artificial intelligence. In order to explain the output, one first has to understand what Wolfram Alpha does in terms of interpretation and handling the request. The engine shall be user friendly. So most database queries are in a form a normal man can't understand (e.g. SQL). So in order to make it usable for everyone, the machine has to understand human language. This proves quite complicated and is mentioned to be one main part of Wolfram Alpha's further development. So how does the search engine interpret the human input? First of all it searches the string for single keywords it knows. These are words like "who" or "age" or "highest". With keywords, the search engine tries to find out what algorithms it should use to compute the user input. The next step is to search the string for data keywords like "mountain", "Munich" or "George Washington". In combination with the found keywords, it links this data and runs the predicted algorithms on it. So the search engine tries to translate human into a computer understandable language. The output varies strongly as there is a huge amount of possible queries.

Now let's look at an example: The query "How many days are there till Christmas 2987" is typed into the textbox of the engine. After clicking the compute-button, the search engine immediately shows how it interpreted the input. It recognized the query in the following representation: days until Christmas Day 2987. The user could have typed these keywords directly into the textbox to receive the same information. The search engine now outputs the result of the query in different forms.



To conclude this part of the paper it is to say that Wolfram Alpha is a new approach to interact with users on the internet. The fact that it is - in contrary to several semantic web applications - usable with human language broadens the user base by a big amount. It also states the sources of the information it provides. So it is somehow more trustworthy than platforms like "Wikipedia". Also the quality and quantity of the output for all input types is precise and sufficient. The quality of graphics and diagrams is good enough to use them as a first overview. Taking all facts in consideration, it is to say that Wolfram Alpha is a well-developed application with much future potential.

Alternative and new technologies to gain knowledge and information

# Google

## Basic Information

Google is now one of the biggest search engines in the world. In 1998 Google went online and since then continued to grow. Right now Google is available in 124 languages. When it went online it had an index of about 25 million documents to search through. Since then this index was steadily increased. In order to increase their index Google uses web-crawler to find new documents and add them to their list. They also try to implement a context based presentation of the findings. For example: If someone often searches topics related to “golf” the sport, then they won’t find a link for a “VW Golf” on top of their findings list when they search for “golf” in Google.

## Technical

Google doesn’t use one big database to store all the information. They instead operate many data centers allotted around the world. This not only makes it hard for Google not to be online, because if one of the centers is offline the other centers will take on their workload. It also has the benefit that it can be used to speed up the search, because whoever goes to their homepage will be directed to the nearest data center. In the data centers Google uses IBM PCs that build a cluster. These PCs use standard components and are therefore very cheap. In addition to this they use their own file system, the Google File System. In this distributed architecture all information is stored multiple times on different machines. The use of such a system also makes it possible to swap PCs without having to shutdown the whole system. The system also designed in a way that many PCs can work parallel on the same problem. This also helps Google to speed up the search. Google uses a reduced version of Red Hat Linux as the standard operating system on their computers.

## MapReduce

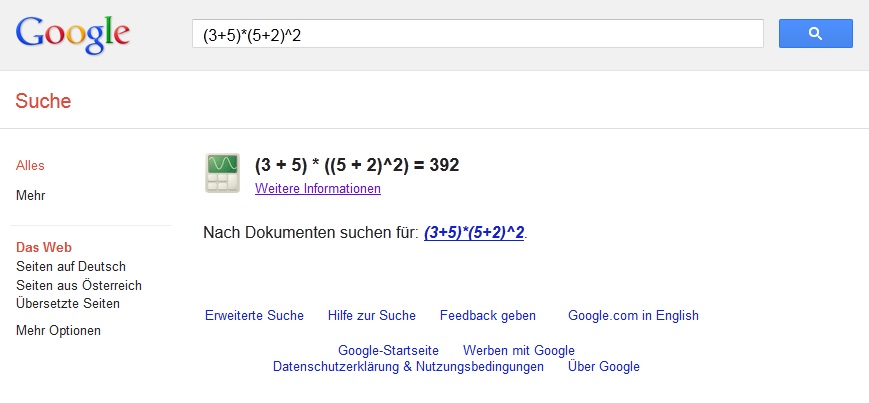
The main aspect of all Google-applications is the MapReduce algorithm. This algorithm was developed by Google to implement automated parallelization in computer clusters. When using this algorithm the programmer only has to implement two hooks.   
The Map-method takes all the input and creates a list of intermediate results in the form of a pair of values. The Reduce-method then takes all the value pairs and creates one for a distinctive value. The main function of MapReduce is to relate all the work that is necessary for the process to be done by the framework. It also controls the distribution of the data, the intermediate results and the collection of the results. It also has methods for error detection.

## Special search functions

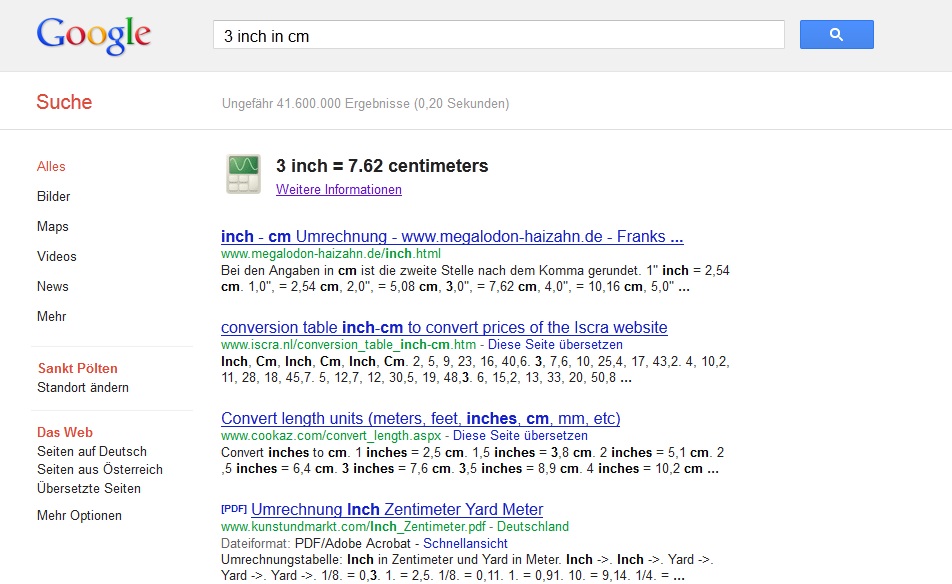
Google has many special search functions. I will just give a short introduction to some of them.

### Calculator

In addition to searching for websites and other things, you can also use the search field of Google as a calculator. If you want to know for example the answer to the following term: (3+5)\*(5+2)^2  
Then just type it into the search filed and press search. After a short moment Goggle will present you with the answer to this term. It’s 392 by the way.



### Unit converter

The unit converter can used to convert units or currencies. This could be use full for comparing prices in different places. To use this function you have to write into the search filed the amount of the unit you want to convert and then write “unit 1” in “unit 2”. For example: 3 inch in cm   
Among the regular search results Google will display the result of the conversion.

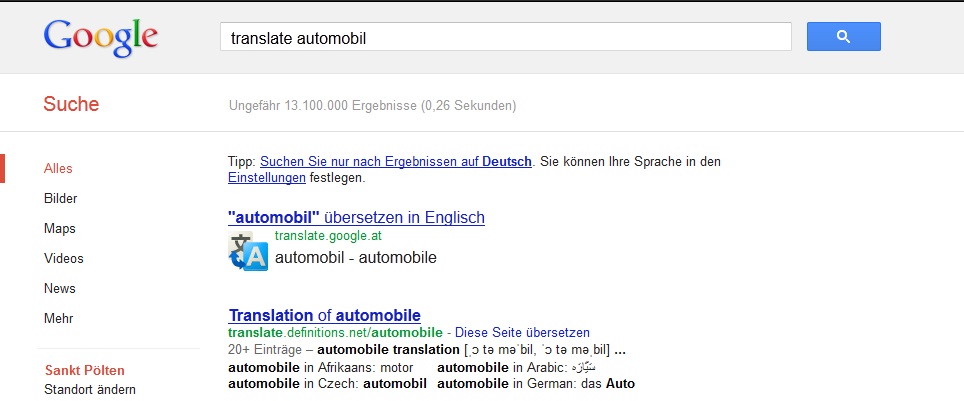
### Spell check

Google can also be used to spell check words. So if you are not sure about how a word is written correctly just type it into the search field of Google and press search. Google will then present you the message: “Results for …?”

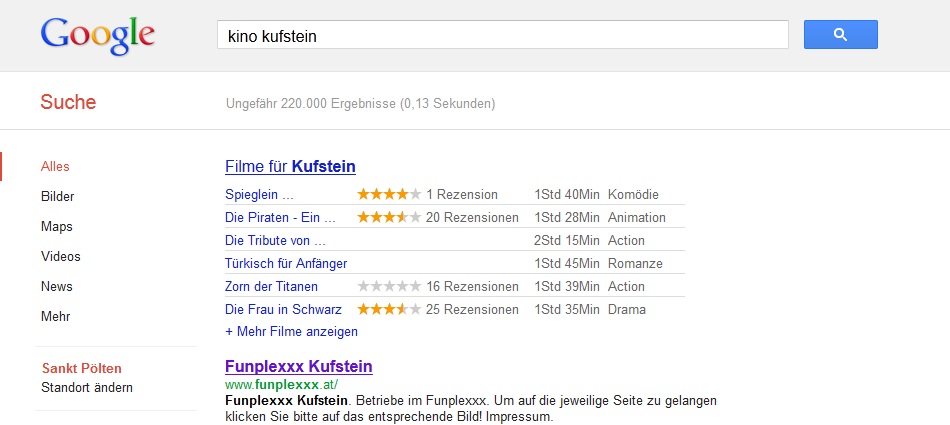
### Address finding

The Google search filed can also be used to search for where a specific address is. So if someone would type in the address of the FH Kufstein Google would show it on a map.

### Translator

Another very interesting function of Google is the translator. With the translator you can not only translate words and sentences, you can also hear how they are spoken. You just have to type “translate …” into the search field and press search. 

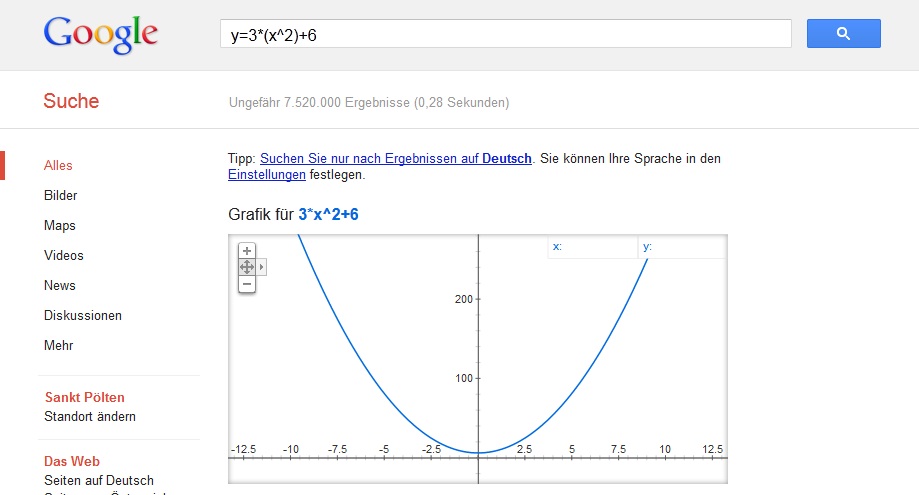
### Cinema plan

If you use Google in Austria or Germany, you can also use it to search for the cinema program. Just type “Kino + location” into the search field. Google will then present you a playlist for the cinemas in the searched location.  


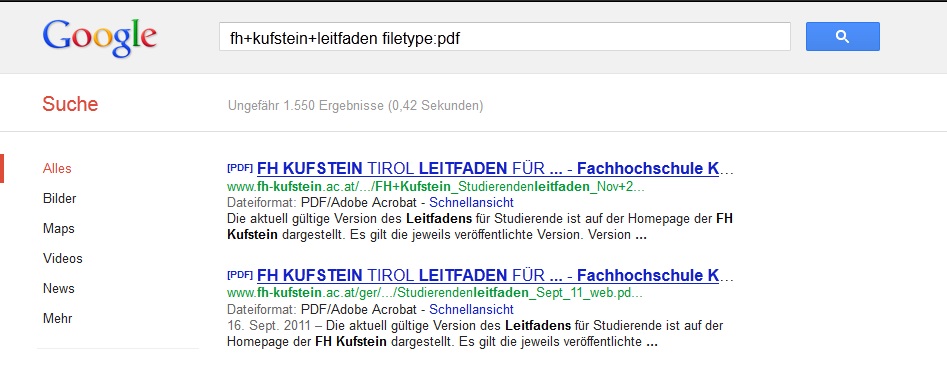
### Weather

One of the most useful functions of Google is the weather forecast. If you type “wetter Kufstein” into the search field Google will prepare a 3-day weather forecast.  


### Graph

Another very useful function of Google is the graph function. It will show you the graph to every mathematical term if it written in the form: “y=k\*x+d”  


## Extended Search Syntax

Google also gives the user the possibility to use extended search syntax to fine tune their search. So it is possible to search only for pdf-files or doc-files. So for example: If you look for the guidelines of the FH Kusftein. Just type ”FH+Kufstein+Leitfaden filetype:pdf” into the search field.  
  
With the filetype extension Google can not only look for pdf- or doc-files but also many other filetypes.

# Greplin

Greplin is the first search engine capable of searching for your private data on the web.[[1]](#footnote-1)

It’s not needed to know where the information is stored. Greplin is able to search Facebook, Gmail, twitter, Salesforce, Basecamp, Dropbox, Evernote, Yammer, tumblr and much more.

There are some requirements to be efficient. Greplin starts being efficient, when many inserts are available, with fewer searches and the per-user cost is low. It inserts up to 5.000 documents per second, and the average size file is between 2 and 4 kilobytes. Greplin uses the amazon cloud. A c1.medium Amazon machine is responsible for up to 80 million documents, and at the same time the machine only has 1.7 gigabyte of RAM. Overall Greplin handles about 50 million documents with 1 gigabyte of RAM which a median search latency of 200 milliseconds.

Per document it takes an average of 27 bytes to be able to search through it. Also with an optimized index, searching through documents, every machine has, it is impossible doing this without swapping.

To decrease the memory usage and therefore increase the efficiency, Greplin had to take certain steps. The user behavior gives needed information, like for users which use the search more often than others, the Index is kept open. For every user there is a separate Search Index, by doing this, Greplin saves memory and improves performance, because there is no more need to search through a whole big index, but just the User’s index. Another advantage is the possibility moving users between servers, figure out their space usage and corrupted indexes only affects one user.

The following picture shows the way how the Ram Index works:

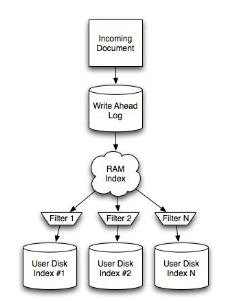


Figure 1: Ram Index

All documents are first added to the RAM index, when the input is a document, the RAM, the currently flushing and their disk index is used. If the RAM index is full, there will be a new RAM index created. Cases like updates and deletions are handled which temporary filters on the disk index.

It often happens that the same document gets indexed multiple times, to detect the duplicates, a filter was added, called Bloom filter.

Greplin is based on the lucene search engine. [[2]](#footnote-2) Lucene itself is based on Java.[[3]](#footnote-3) If users misspell a word, a separate search index is needed, in terms of privacy, performance and quality. Greplin came up with a solution for that dilemma; they use a Spell Checker class, which is a two-tiered spellchecking system. At first a global index is build, based on dictionary words, then so called user indices filter out words which are already present in the global index. Words which occur under a certain frequency are also filtered out by the user indices. The result of these measures is a depreciation of nearly 80% lower size of the indices (for English spelling users).

There are more problems which occurred during the developing and the productivity process of Greplin. For example a simple underscore or dashes. Greplin indexes Dropbox and email, the query {busy\_cat} is matching an email containing “A busy woman named Cataleya”. A solution for this problem was found, turn tokens connected by dashes and underscores into phrase queries. Another less typical problem like #hashtags and @handles vs. addresses (Apartment #120) and mail addresses also occurred during the process, the solution therefore: keep meaningful punctuation and ignore the rest. The query {1200 Orleon Boulevard #120} and {1200 Orleon Boulevard 120} will both have the same result. The Greplin team forked the StandardTokenizer JFlex and built their own lexer, also which this workaround, query parsing still requires hacks. The team still works on existing problems.

Greplin is a relatively new service, that’s the reason it doesn’t get mentioned very often in the web. But Greplin does not deserve to get lost in the web, like many other startups. As already mentioned before, it’s the first service for searching like this, but it speeds up the search for information in your private data, like never seen before.

The following pictures will show how Greplin works:



After the account creation and connecting the services (like Facebook, Twitter,…) it just takes a few minutes, depending on the amount of information, and Greplin is ready for the search.



Typing just a few letters and you already get all the results with the letter’s “aw”. Just one more letter and you get “awe” and already the information searched for.

There are already Greplin apps for mobile devices.

If someone wants to follow the development or share solutions, they can take a look at github.com/Greplin.[[4]](#footnote-4)

# Conclusion

As we can see, there is no “all in one” solution. Every search task requires another search engine, therefore another way to search. That could be the reason that so many different search engines exist. At first Google was just able to search through html pages, but they expanded their abilities, already mentioned before in this document. Google kept growing and growing and is one of the most successful search engines.[[5]](#footnote-5)

When there are already too many products with equal or nearly equal services, a way to success could be the specialization. Wolfram Alpha and Greplin are specialized search engines. The document already described how Wolfram Alpha works, it tries to give logical solutions for search queries. On the other hand, there is Greplin, which doesn’t search through the web, like Google and Wolfram Alpha. Greplin is more for these niches, the first two are not able to reach, the private cloud, and of course it works a little bit different.

It all depends what and how you want to look for information. At least it can be said, that all three presented search engines have their areas in which they are strong.

The authors of this document expanded their minds and their possibilities by using these search engines.

1. Fc. (Böhm, 2012) [↑](#footnote-ref-1)
2. Fc. (Walker & Manek, 2011) [↑](#footnote-ref-2)
3. Fc. (Apache Software Foundation) [↑](#footnote-ref-3)
4. Fc. (Cvet) [↑](#footnote-ref-4)
5. Fc. (PCWelt) [↑](#footnote-ref-5)