Cryptic Crossword Generator Powered by a Search Engine

Karl Buckland
Computing Laboratory
University of Kent
Canterbury, Kent
UK, CT2 7NF
kab22@kent.ac.uk

Abstract

Cryptic crossword clues are designed to be deceptive and misleading and as such are constructed by people. Although there are some basic software solutions to this problem, they rely on a 'cookie-cutter' approach to producing cryptic crossword clues. What if it were possible to use an automatically-updating database of information, such as a search engine, to power the generation of clues and how effective would this method be at producing clues compared to other methods?

This paper details how best to use a search engine in this manner, the methods devised produce good results and the suitability of the results compared to human created clues.

1. Introduction

Cryptic crosswords are a more complicated version of normal crosswords. Instead of simply providing 'straight' or 'quick' clues, which are usually just definitions of the answer, each cryptic crossword clue is a word puzzle in and of itself. The surface reading is almost always never anything to do with the answer, therefore solving the clue relies on the reader being able to figure out how to read the clue, and therefore find the answer.

Most clues are made up of two parts; a definition or synonym and a wordplay description of the answer. Frequently this involves anagrams, hidden words or playing on words. The resulting clue is therefore quite misleading as it has nothing that obviously connects it to the answer.

Constructing cryptic crossword clues requires a number of abilities. One must be able to:

- spell words accurately
- provide short descriptions of words (such as 'short' or 'quick' clue types)
- produce anagrams
- use 'indicator' words
- arrange the answers into a crossword grid

The above components must also be used in the correct manner for each clue type, of which there are many.

1.1 Cryptic crossword clue types

As part of this introduction I need to quickly outline the different clue types that one finds in a cryptic crossword, as this knowledge is necessary to understand the challenges in constructing these clue types and therefore this will directly inform the reader of the appropriateness of my solutions.

'Pure cryptic' (also known as 'double entendre') clues are a simple play on words, but usually not with the dictionary meaning. Instead they rely on alternative uses of a word. For example: 'The flower of London (6)', the answer to which is 'The Thames', as in the flow-er of London. This type of clue requires a lot of explicit knowledge, such as the fact that a word can be suffixed with 'er', even though the resulting word isn't a real word in the sense in which it was defined, and that there is a river in London called the Thames. Most pure cryptic clues use general knowledge, or wordplay in this fashion.

'Double definition' clues contain two different definitions of the same word and usually each definition is of a different meaning. For example: 'Not seeing window covering (5)', the answer to which is 'Blind', as in 'to be blind' (not being able to see) and 'window blind', as in a blind that covers a window.

'Hidden word' clues are when the answer appears in the clue, hidden in one or more words. This clue type usually also contains a synonym or straight clue. For example: 'Unsound ermine descends to disobey (9)',...
the answer to which is 'undermine', which can be found from the end of 'Unsound', into 'ermine' and is clued directly by the straight clue 'to disobey'.

'Reversal' clues are when the answer gets turned around to make another word, which is again clued. Like many of the other clues, this clue type also contains a synonym or straight clue. For example: 'revolutionary beer befitting a supreme ruler (5)', the answer to which is 'regal', as 'regal' backwards is 'lager', or 'beer'. The phrase 'befitting a supreme ruler' is a straight clue for 'regal' and the word 'revolutionary' is a reversal indicator.

'Charade' clues are formed by joining together individually clued words to make a larger word. For example: 'outlaw leader managing money (7)' is 'Banking'. 'Ban' is 'outlaw', 'king' is 'leader and a straight clue for 'Banking' is 'managing money'.

'Anagram' clues are generally the most popular and consist of an anagram of the answer and a straight clue. For example: 'Chaperone shredded corset' is 'Escort'. 'Corset' is an anagram for 'escort', 'chaperone' is a synonym of 'escort' and 'shredded' indicates that 'corset' is an anagram.

'Homophone' clues are made from words that sound the same but are spelled differently and have different meanings. For example: 'We hear twins shave (4)' is 'Pare', because 'shave' is another word for 'pare' and 'twins' is a 'pair' as in 'pare'.

The final type of clue is the 'combination' clue, which can be a combination of any of the above, often in any order and with multiple indicator words as a result.

1.2 A short overview

Using a search engine to inform the construction of crossword clues involves a number of distinct steps and therefore my research has been focused in several areas.

Firstly, one must ascertain if such a complicated use of the English language can be artificially generated with the available tools. Section 4 deals primarily with answering this question, but doesn't go as far as detailing the necessary steps required to construct the clues.

Secondly, different methods of utilising the search engine must be researched and the results of those methods used to discover which methods are best at informing crossword clue generation. This issue is outlined in section 5.

Lastly, section 6 combines the information from the above two areas, to discover how the search methods can be used directly on the different clue types and what results can be expected.

2. Background

2.1 Crossword generation

Crossword generation has long been researched and worked on so that today commercial generators such as 'Crossword Compiler'[1] exist which are capable of laying out answers into a grid and producing 'straight and quick' clues. These are preprepared or generated from preprepared sources. However automated compilers are rare and most crossword generation services offer hand-made crosswords. Perhaps this is because crossword setters (people who create crosswords) and cryptic crossword setters especially are challenged to produce original clues. This is particularly true of the popular crosswords found in national newspapers.

While cryptic crosswords may still be put together by hand, it is reasonable to assume that expert setters have resources available to them such as lists of indicators, definitions and synonyms which are easy to acquire. However, it is not known if the clue generation process itself is informed or inspired. Perhaps it is just enough to see the answer and try to create a clue.

Considering the wordplay and innovative elements that make up cryptic crossword clues, it is surprising that I could find no programs capable of creating them. There are certainly plenty of solvers, which appear capable of deconstructing clues such as 'Wordplay Wizard'[2] and 'Crossword Maestro'[3] and are therefore capable of understanding the various components that make up the clues. It is likely therefore, that the lack of a database containing all of the necessary components (especially wordplay and general knowledge elements) and possibly the expectation of little demand, have led to the work required for such a program/service to be seen as unworthy of the potential gain. Evaluating the success of cryptic crossword solvers is difficult, but Crossword Maestro claims a success rate of 73% given the clue alone.

2.2 Sentence construction

Sentence construction traditionally applies grammatical rules to existing information. Research in this area has mainly focused on natural language and story generation and it is widely recognised that the resulting text is significantly different to compose for each.

The biggest challenge in generating convincing prose seems to be that of context. The words used in a sentence, paragraph or entire article/story need to be
related to the subject at hand. This is especially important in story generation. This area may be easier to solve in cryptic crossword clues, as only small amounts of material are required to produce one clue and the clues themselves do not need to relate to each other.

As well as context comes the issue of suitability and the application of real-world knowledge. For example, what is a shirt, how is it used, what is it made of and therefore how could it be used unconventionally? Yet another issue is that of related concepts.

Wordnet (see section 4.1) has been used in the past on many language-related projects to enable the projects in question to easily produce related concepts. One such project, 'Wordnet Enhanced Automatic Crossword Generation'\cite{4}, uses Wordnet to produce straight cryptic crossword clues, where each crossword can have a specific theme, taking advantage of Wordnet's ability to link words to each other by meaning. This is possible because Wordnet's databases are hand-collated, but it should be telling enough regarding Wordnet's suitability that an entire project has been dedicated to producing good straight clues from the wordnet database; a great deal of processing is still required.

When initiating this project I clearly wanted to research the use of a search engine with regards to generating crossword clues and therefore I have made a conscious effort to avoid using more traditional methods, so that this area of research can stand alone. As soon as traditional methods are mixed with search engine use, it becomes much harder to ascertain the usefulness of both and how each has contributed to the results seen.

### 2.3 Generating or verifying content using a search engine

The APIs provided by search engines (see section 4.2) are fairly new with Google's API being established in April 2002\cite{5} and the Yahoo API established in March 2001\cite{6}. However both of these services suffer from restrictions on the amount of use any one user, or application can perform. The main focus for both Google and Yahoo has been on providing web-centered services to allow website owners to produce new content and services using the APIs, as shown by Google removing support for their web API and instead pushing their 'web toolkit' and AJAX solutions\cite{7}. These factors may go some way to explaining why there has been little notable research utilising these services.

The APIs provide the same information one can find when searching through a web browser. The response to a query contains 10 results, where each result has the web page's title and URL, and a short 'snippet' describing the page. The result also contains a number representing the number of total results. It is also possible to request additional pages from the search. Most commonly-used search string operators can also be used, such as using quotes to search for a specific fragment, or 'plus' and 'minus' signs to specify the weighting of search terms. Most advanced search methods can be used too, such as 'site:x', to search for results in one site.

The nature of a search engine requires that it is optimised to help people find what they were searching for. This is not optimal when trying to extract fragments of language or decide what the best course of action is when, for example, constructing a crossword clue. Indeed, the search engine needs to have no actual understanding of how or why the results relate to the query (on a meaningful level, such as different uses of the same word). Instead search engines use the tried and tested assumption of sorting by popularity. If, for example, the user receives a result which is unexpected, then he/she simply refines the search parameters. Therefore it can be assumed that search engines themselves have been created with assumptions in mind. An optimal source of information for producing crossword clues would be to use a specially-constructed database, but of course this solution would require manual, intelligent updating, whereas a search engine is automatically updated from an almost limitless source (the Internet).

With most content received from search results being fragments, and with the search engines seemingly unaware of context, the content from the search engine itself cannot easily be used to generate new content. After all, the search engine is there to help users find content, not to provide that content to them directly. Many of the results contain brand names, improper uses of words, improper results or other unexpected factors which damage the usefulness of the results when generating new content. One important factor is the large number of Internet stores and their affect on results; a common phrase or unsuspicious sentence can often return a number of commercial results. The only reliable way to generate new content is to examine the content that each result represents.

The first few pages of results for a query are often very similar, or the same results may appear to come up again and again. This appears to happen as content is published, discovered and then linked to or copied multiple times. This again reduces the reliability of search engine results when trying to generate new content, as any content produced could be skewed by
unforeseeable content being ranked highly on particular search terms.

3. Aims

The overall aim of this project is to construct a working cryptic crossword generator, powered by a search engine. However it is difficult to measure, in absolute terms, the effectiveness of the implementation of such a project when dealing with subjective results. It is therefore necessary for me to quantify the success of the project through qualitative means and for the reader to decide if the reasoning contained within this document is the same as their own.

With this in mind and with the overall focus of the project being to research the target area in detail, I will phrase more detailed aims as questions or leave them a little open-ended in order for me to provide detailed answers at the end of this report.

The first aim is to discover the suitability of a search engine for informing the generation of cryptic crossword clues. Can a search engine be used effectively in this manner?

The second aim is to prove that the clues created are positively influenced when using a search engine. I will detail the methods I have used to most effectively create the clues and demonstrate the difference that these methods bring.

The last aim is to create a functional cryptic crossword generator, capable of producing clues from answers fed into it, and generating a crossword layout.

4. Creating the different clue types

4.1 The tools available

Section 1 contains a short list of the abilities that one must have in order to construct cryptic crossword clues and here I will briefly outline the tools I have used in order to supply those abilities.

In order to spell words correctly, this project uses DICT[9], which is a dictionary protocol for which one can obtain a server and client and is capable of supporting multiple dictionaries. By default my development system uses the GCIDE[9] dictionary.

To produce synonyms and 'short' or 'quick' clues this project uses Wordnet[10], which is a large lexical database, capable of using thesaurus data to produce synonyms, definitions of words grouped by meaning (senses) and much more. The thesaurus used in this project is the Moby Thesaurus, described as the 'largest and most comprehensive thesaurus data source in English available for commercial use'[11].

The synonyms that Wordnet can produce with Moby are extensive, sometimes negatively so when the synonyms listed are of rarely-used meanings of a particular word, or are very loosely connected. Unfortunately, this has been an obstacle that I have not been able to overcome and supports why a crossword-specific database may be more appropriate.

Wordnet rarely produces descriptions of words which could be used as 'short' or 'quick' clues, as most descriptions are long, formed of multiple descriptive words or clauses, or use the word being described (or a variant of it). Therefore significant processing has had to take place on these definitions in order to make use of them in cryptic crossword clues. Due to the fact that most words have multiple definitions and in order to fully research whether a search engine is beneficial in improving the clues, I have tried to do little analysis on the fragments within the program. Instead the fragments are cut up into small chunks (or discarded if they are too long or contain the word they are describing) in order to produce short, concise fragments.

In order to produce anagrams, the project uses WordPlay[12], a small, fast anagram generation program written in C. Once the anagrams are produced, the program ensures that the resulting anagrams exist in the dictionary in order to be useful to the program.

Indicator words (words in the clue which indicate which type it is) have been compiled by hand from existing crossword clues and various websites which list them. The program implements three different structures for anagram clue types and therefore uses three different types of indicator words in order to prevent anagram clue types generated by the program being recognisable due to their structure.

4.2 The feasibility of producing clue types

I have already established that cryptic crosswords are complicated, that their creations requires a variety of components and that the ability to create these clues is often perceived as a talent and is paid for accordingly. It is therefore arrogant to assume that a computer can create clues with the same detail and variety, especially as some of the clue types contain general knowledge, or make assumptions with wordplay that require knowledge of the world; something that a computer cannot emulate without intelligence or an extremely comprehensive database. However, some of the clue types are good clues, not through their construction, but simply because they are not what they first appear to be. There is no reason why these clue types cannot be created artificially, even if it is just as inspiration for a human setter.
By referring to the clue types established in section 1.1, one can determine that 'pure cryptic' clues are too complicated to construct without a dedicated database of information on which to draw. Seeing as the focus of this research is to produce convincing clues by using a search engine, I believe that using general knowledge and original wordplay could easily make this aspect more complicated and thus dilute the relevance of any search engine results. Building a database of such information would also require significant time and research, arguably enough to justify another project entirely. For these reasons 'pure cryptic' clues will not be attempted.

Finding a suitable source of homophones in an easily usable form has proven to be difficult, as I could find no established accessible database. Also, while not as complicated as 'pure cryptic' clues, homophone clues offer no individual challenges over the other type of clues when it comes to their construction and the use of a search engine to improve them.

All of the other clue types can be constructed using already available tools and components. Double definition, hidden word, reversal, charade and anagram clues are all possible and offer difference challenges for this research to tackle.

5. Search methods

5.1 An overview of search capabilities

Both Google and Yahoo APIs offer similar services to developers, with both offering interfaces to their respective services to a variety of languages, including Java. Indeed, the actual services are almost identical, except that Google uses SOAP and Yahoo uses XML. Unfortunately, the 1000 search limit a day imposed by Google and intermittent problems with the performance of the service forced me to switch over to use Yahoo instead, but the result is that the project is capable of supporting either API with only minimal changes.

In the introduction I covered the parts of which the search results consist; a title, URL and snippet is provided for each result. I will now briefly explore the usefulness of these parts.

The title is not particularly useful in developing methods as it may or may not contain the search criteria and may or may not actually be related to the content of the page. Many web pages still bear the title “Untitled Document” (this query returns 33.6 million results at the time of writing) and many titles have only the name of the website, the section or a mixture which may or may not be useful. Well named web page titles would arguably be very useful, as they would contain short, concise descriptions of the page content, potentially making them good examples for crossword clues, but as this is not the case for many page titles this theory cannot be adequately explored.

The snippet contains the areas of the web page in question that contain the search query items. This means that the snippets are sometimes one fragment from the source, or sometimes multiple fragments. Frequently, this means that there is not enough of the surrounding text to make sense of the fragments in any context or to draw any solid conclusions from the snippets themselves for the purposes of constructing a cryptic crossword clue.

The URL is perhaps the most useful part as it contains the elements from the search query, in context, and could directly inform the application. Unfortunately, this requires downloading the content of the URL, which takes time, and then processing that content which takes further time. For another application, this process could be extremely useful, but as I will detail in the coming sections, generating enough information from such content would require a great deal of time and processing power.

Developing search methods for this project has required exploring various aspects of the search engines that I have not used regularly before, such as more advanced search features. Advanced features such as searching:

- for content on a specific domain
- with/without particular words
- for results in a particular file format
- where the search terms appear in different places (links to page, the page content, the title)
- for pages that are 'similar' to another
- for pages that link to a given page

These advanced features have not been exploited directly because they offered no advantages over other methods. The snippets provided with each search result have the potential to be the most useful without loading any additional content, but they generally contain information from various points in the source, so they cannot be used to draw solid conclusions.

I have already mentioned that Google has a 1000 search a day limit in place on its service. Yahoo has an undisclosed limit and will penalise you if the limit is broken, but so far this limit has not been an issue. Regardless of the technicalities of breaking the limit, searching takes time as the request has to be made to the service and a response generated and sent back. Searching is therefore quite an expensive process for an application, not just because nothing can be done whilst waiting for the response.
It is therefore necessary to use as few searches as possible, as wisely as possible. This behaviour is sought regardless of the above requirements in a computer science context, but in this case it is especially important.

5.2 The 'word frequency' method

The first method, 'word frequency', measures the frequency of words directly before or after another word or sentence fragment. It searches for the word/fragment and builds up a frequency table of results. My research has shown that, given enough searches, and programmed with a reasonable margin for accepting a result, this method can produce reasonable results. However, how long should one search for?

By being more specific in search requests, one may expect to narrow the scope of the search results and produce a faster answer, but this often has the effect of substantially reducing the number of results and 'scraping the barrel' when it comes to the quality of those results.

Clearly a cut-off point, such as a page limit would be necessary, especially on the more vague searches which could continue for tens or even hundreds of pages, especially when faced with hundreds of thousands or even millions of results. If one were to introduce a page limit when performing this search method, then it would have the effect of stopping the more vague searches before they produced a result, even though they are more likely to find a good result.

The biggest problem with the word frequency method is that of failure. Supposing there is a page limit, a failure can only occur when all of those pages, and thus that many searches have been expended. At the end of this process the only benefit is the knowledge that the particular search could not generate a solution. Failure, in this case, would use up more resources than a successful conclusion.

A major criticism of this method is deciding what to do with the result when it is obtained. Another method has to be employed to ensure that the fragment which is found to be a good solution, can be used with another fragment. A classic example of this would be the challenge in joining two sentence fragments together; using this method to solve that challenge would take even longer.

5.3 The 'commonly associated words' method

The second method 'commonly associated words' picks out words that occur frequently in the results in order to find words commonly associated with the words in the search term. This method also builds a frequency table of the results and is generally improved by the number of words used, up to a point.

This method was originally designed to create synonyms for existing words, until WordNet was discovered. It is possible that this method could instead be used to replace words from existing fragments in order to generate more possibilities when generating content. However, considering the dangers of using content directly mined from search engines, a method such as that could compound errors by using search terms which could have been inappropriately mined.

It is clear that this method is only slightly different from the word frequency method, although it does produce very different results, however the same disadvantages of that method also apply in this case; this method has the potential to deliver good results, but only after a costly time investment.

5.4 The 'popularity contest' method

The last method is the 'popularity contest' which does not look at the search results at all, instead it looks at the number of results, which itself is returned as a number. By comparing similar strings and using the one with the most results, this method can reliably produce understandable fragments of sentences.

This method, unlike the previous two, only requires one search to be performed for each string that needs to be rated. Even allowing multiple strings to be compared can mean that the total number of searches required is quite low to produce an adequate result. By using the total number of results as the measure of acceptability, this method takes advantage of a search engine's ability to sort and rank data, and uses that data to rank the search queries for itself.

Details of how the popularity contest is used to improve clues can be found in section 6, but here I will briefly outline why this method is useful by using anagram selection as an example.

Table 1 shows anagrams of the word 'thunder' and the number of results each anagram receives when searched for in quotes (the entire string) on Yahoo! search.
The anagrams in Table 1 have been ranked roughly according to their popularity on the Internet in general and while it is difficult to see why some of the anagrams rate higher than others, the anagrams that score highest are certainly likely to be used more commonly.

The obvious downside to the popularity contest method is that, one must already have content to rate, whereas the other two methods will generate new content. However, as I've already established, a great deal of the content mined from search engine results directly is inappropriate and therefore generating new content is particularly challenging.

Assuming that content is already present, the popularity contest can easily be used to produce results on word frequency and commonly associated words, much faster than results could be expected from either of the other two methods.

The popularity contest method is certainly the most reliable and the least expensive method to use, assuming that the number of options you have to try is small or can be selected to be small.

6. Utilising the search methods

The most successful use of the popularity contest is in the anagram clue type. As the example in Table 1 from section 5.4 shows, the popularity contest is naturally good at selecting the best option from a list.

The popularity contest can only be used to select the most popular string from a list, but the strings themselves can of course be generated so that the result performs a particular decision.

When constructing the double definition clues the program needs to work out the best way of linking the two individual straight clues together. To do this, several possible joining words are selected, such as 'and' and 'to' and then the possible fragments of the sentences that would be generated using those joining words are created. In order to produce significant search results, only the last word from the first fragment and the first word from the second fragment are included.

For example, to produce a clue for the answer 'decide', the following short clues are selected:

- bring to an end
- settle conclusively

From these, and the joining words above, the possible search queries are generated. Table 2 shows the queries and the number of results each query receives.

<table>
<thead>
<tr>
<th>Search queries</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>end and</td>
<td>25,000,000</td>
</tr>
<tr>
<td>and settle</td>
<td>2,410,000</td>
</tr>
<tr>
<td>end to</td>
<td>46,200,000</td>
</tr>
<tr>
<td>to settle</td>
<td>24,600,000</td>
</tr>
<tr>
<td>end settle</td>
<td>719</td>
</tr>
</tbody>
</table>

The queries 'end to' and 'to settle' receive the highest number of results, therefore the final clue, 'bring to an end to settle conclusively' is produced. This is clearly better than 'bring to an end settle conclusively'.

The charades, hidden words and reversal clues use this method to decide on the order of elements and the suitability on the short clues to the other elements.

To produce a reversal clue for the answer 'part' requires two short series of searches. The following short clues are selected:

- go one's own away
- go different ways
- in some degree
- assets belonging to
- due to

The popularity contest method selects 'due to' as the most popular clue. Unfortunately this is not necessarily because it is more appropriate, but as it is a shorter statement of commonly used words, it is much more popular.

Now the program has to decide on the order for the words in the clue. The two possibilities are:
Reversed due to oral cavity
oral cavity due to reversed

Table 3. Reversal clue structure popularity

<table>
<thead>
<tr>
<th>Search queries</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>cavity due</td>
<td>5,130</td>
</tr>
<tr>
<td>to oral</td>
<td>1,170,000</td>
</tr>
</tbody>
</table>

In this case, the more sensible sentence has been selected. This method is admittedly quite risky, as it only tests the popularity of a very small fragment of the sentence. Therefore the key to making this clue type work is in keeping the whole clue as short as possible.

6.1 Search query preparation

Whilst the popularity contest search method is useful for improving clues, it is only as good as the content supplied and therefore it is necessary to prepare data for searching so as to make best use of the results. During the course of researching and constructing the cryptic crossword generator I have also tried, wherever possible, to select the most appropriate data before performing a search. The benefits to this are that less searching is required, or alternatively, that better queries can only lead to better results. When performing searches on randomly selected data, it is possible that the randomly-chosen data represents a poor sample. The search methods may pick the best result, but the best of a poor bunch is still a poor result.

One area in particular that showed huge improvement was the hidden words algorithm, which works by finding words of a reasonable length, within a given word. For each hidden word, there is a prefix and/or a suffix, containing the remaining letters of the original word. Originally I decided that selected the longest hidden word, preferably with both prefix and a suffix was best, as this would hide the word well in most cases. Unfortunately this frequently meant that the prefix and suffix were one character only, and the resulting random words generated from them were generally entirely unrelated, which made the sentences seem very random in their construction. Selecting five random words and using the popularity contest method often failed because of the poor list from which the search method had to choose.

The solution was to try to find a balance between a long hidden word and a prefix or suffix (or both) that was not too short. The longer the prefix or suffix, the smaller the list of possible words to which they could belong became and the higher the likelihood of creating more natural language in the clue. To achieve this, scores are assigned to the various hidden word options. The scores are calculated as follows:

For three-part clues:
- First part = number of letters x 2
- Middle part = number of letters x 0.5
- Last part = number of letters x 2
- Total = sum of the previous three sums

For two-part clues:
- Total = number of letters in first part x number of letters in second part

This scoring method favours a well-rounded three-part clue, over a balanced two-parter, over a badly rounded three-parter. The result is that fewer words are generated from the resulting prefixes and suffixes and better words are chosen as a result by the search engine methods.

6.2 Search concerns

As I have previously discussed, the search engine methods that this project uses aim to use the smallest number of searches possible to produce a result. It is therefore necessary to look at how long the various clues take to complete and how many search queries are involved in the creation of clues and crosswords. Most of the clues aim to search between 0 and 10 times depending on the data available, although most search about 5 times on average. As I have previously explained, the searching methods for each clue type have concentrated on a different area, so ideally, each clue type could benefit from the search methods available on the other clue types. This would increase the number of searches, as well as the quality of the clues.

One casualty of the restriction on the number of searches has been the flexibility of the clue types and the likelihood of producing a clue at all. Producing straight clues from the Wordnet is often quite challenging due to the sentence fragments that Wordnet has in its database. In order to have the best chance of producing a sensible clue, the cryptic crossword generator is quite harsh on the definitions it uses. Once this is coupled with the reality that not all words have anagrams of them, or cannot be reversed, or hidden in larger words, then sometimes only one or two clue types can use any single word. This can only really be improved by improving the sources for all of the mined data.
7. Conclusions

7.1 Comparisons to traditional methods

The nature of a project such as this, attempting to marry topics as unconventional as cryptic crosswords and search engines, has meant that there has been nothing to directly draw upon for background research and therefore it is difficult to draw a direct comparison between this project and another. As I briefly mentioned when discussing the aims for the project, this means that it is particularly difficult to discern whether or not the project has been successful.

The research I have undertaken shows that search engines can be used to construct original content, or inform the construction of content based on existing assets, but that the time and resources required to do so perfectly are currently too extreme. Instead, my project can produce some good clues, some acceptable clues and some extremely bad clues, where the necessary parts do not quite come together.

If it were possible to conduct more searches, then this project could have taken the opposite approach, aiming to be as accurate as possible, but taking a long period of time. It is likely that some interesting findings would result of such an approach.

Traditional sentence construction is far more advanced and detailed and produces better results than the methods I have outlined in this report, but traditional methods are also firmly rooted to the sources of information on which they are dependent. Perhaps an interesting combination would be to use a search engine to provide new content, or discover the likely uses for content without human intervention, but to leave the actual processing of language to tried and tested methods.

7.2 Uses for this application

As it stands, this application is useful for inspiration if nothing else. It frequently uses meanings of words which are not obviously apparent, but can also produce combinations of words which are unexpected and therefore well suited to cryptic crossword clues.

This application can also produce clues at a rate of roughly 3-10 seconds each, which seems reasonable when several search engine queries are involved.

The application is also quite scalable, as it makes adding new clue types easy and also provides a large library of commonly-used functions on strings and for performing the popularity contest.

7.3 Future possibilities

The future of this project and similar work is quite open-ended. There is a great deal of room for improvement on the application, such as adding additional clue types, using more traditional sentence construction methods with the search engine methods, or improving the search engine methods and programming over time with more research.

Alternatively, there are many projects in a similar vein which could be attempted, such as aggregating content from the web.

The most interesting possibility would be the creation and use of a database which allows computer programs to understand how to use words in context and explore the links between words, much like Wordnet permits on a basic level. One such project, giving computers 'common sense', is OpenMind Commonsense[13] however it is very much in the early stages.

As an aside, it would also be interesting to see what effect the final output of the application would have were that information to be published on the net and then indexed by search engines. Should the application be used widely, it may be possible for feedback into the application to start affecting results. If a project utilising search engine results ever becomes large enough to suffer from this problem, then the results themselves may be worthy of study.

7.4 Fulfilling the requirements

I have already answered the question of the suitability of a search engine for informing the generation of cryptic crosswords: A search engine is not good for actually generating content, but is instead quite good at informing the user if what they have is good or not. This makes sense from the perspective of a normal search engine user, who will tailor his/her search queries in order to get the results they want. It's arguable that receiving no results is useful information for a normal user, it is certainly useful for a computer program.

Section 6 details how the search engine queries have improved the clues over simply randomly joining text and while traditional methods can construct sentences, they might have trouble trying to ascertain which of a series of anagrams is the most appropriate. The search engine methods I propose can do this with ease, however they have difficulties with sentence construction, which traditional methods have mastered.

Although I have not touched upon it in this report, the application does work fully, has a GUI and does lay out the crosswords in a grid and provide the clues. The crossword grid can be filled in and the answers checked, or the grid can be printed out. More
information on the application itself can be found in the application documentation.

Overall I feel that the research has been successful in establishing the potential use of a search engine for this kind of work. It is partially useful, at least for inspiration, but would not be ready for anything more than interest or research without significant development and better support from the search engine companies themselves.

8. Acknowledgements

I would like to thank Colin Johnson for his continued support and ideas throughout the creation of the cryptic crossword generator.

9. Bibliography

[1] Crossword Compiler
http://www.crossword-compiler.com/

http://www.crosswordman.com/wiz.html

[3] Crossword Maestro
http://www.crosswordmaestro.com/


[5] Google releases Web APIs, Slashdot.org
http://developers.slashdot.org/article.pl?sid=02/04/12/1128210&mode=thread&tid=156

[6] Yahoo Debuts Search APIs
http://developer.slashdot.org/article.pl?sid=05/03/01/148220&tid=95&tid=8

[7] Google SOAP Search API Homepage
http://code.google.com/apis/soapsearch/

[8] The DICT Development Group, dict.org
http://www.dict.org/bin/Dict

[9] GCIDE, Wikipedia.org
http://en.wikipedia.org/wiki/GCIDE

http://wordnet.princeton.edu/

[11] Moby Thesaurus, Department of Computer Science, University of Sheffield
http://www.dcs.shef.ac.uk/research/ilash/Moby/mthes.html

http://hsymovies.com/static_subpages/personal/wordplay/

[13] OpenMind Commonsense
http://commonsense.media.mit.edu/cgi-bin/search.cgi