JIS

Google stemming mechanisms

Ahmet Uyar

Faculty of Engineering, Mersin University, Turkey

Abstract.

In this study we investigated the stemming mechanisms of Google. We used its web interface and submitted many queries via a program. Stemming is the process of correlating morphologically similar words with one another. Search engines use stemming to match documents having one form of a word with queries having another form of the same word. We investigated the stemming mechanism of Google for three classes of words: singulars/plurals, combined words, and verbs with many postfixes. Our results indicate that Google uses a document-based algorithm for stemming. It evaluates each document separately and makes a decision to index or not for the conflated forms of the words it has. It indexes documents only for word forms that are semantically strongly correlated. While it indexes documents for singulars and plurals frequently, it rarely indexes documents for word forms with the postfixes of -able or -tively.

Keywords: Google; search engines; stemming

1. Introduction

Search engines are one of the central components of the web and they play a crucial role in information discovery. They constantly crawl the web and perform the searches on their databases. They try to identify the most relevant documents for the given queries and return the top results to the users. They are extremely complex engineering products and use many techniques to provide the best services. One of the techniques search engines employ is called 'stemming'. In this study, we investigate the stemming mechanisms of the Google search engine.

Stemming is the process of correlating morphologically similar words with one another. It is based on the assumption that these words share similar meanings. Stemming algorithms determine the set of words with the same roots to form equivalence classes. The most widely used stemming technique is suffix stripping which is the process of removing suffixes from words iteratively until the stem is reached. This method was first proposed by Lovins in 1968 [1] and many variations have subsequently been developed. Today the Porter stemming algorithm [2] is the most commonly used method. Dictionary-based stemming is another common technique [3]. For this type of stemming, equivalence classes are determined in advance and saved in tables. The

Correspondence to: Ahmet Uyar, Mersin Universitesi Çiftlikköy Merkez Kampüsü, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Mezitli/Mersin, Turkey. Email: ahmetuyar@gmail.com

algorithm checks the given word in the tables and relates it to its group. It is highly accurate for the words that exist in the tables but it cannot work for the words that are not included in the dictionary. There are also some other techniques such as the successor variety method [4]. A good overview of stemming can be found in [5].

Search engines use stemming to match documents that do not contain the words in the query but have the conflated form of a word in the query. This increases the number of documents that can be matched significantly. However, matching more documents may lead to less accurate results. While stemming increases the recall, it may decrease the precision. Therefore, search engines should be careful not to relate semantically unrelated but morphologically similar words. Google [6] and Live Search [7], but not Yahoo [8], use stemming for English.

The use of stemming in search engines comprises two steps. The first is to determine the equivalence classes for words. This is done by using one of the stemming algorithms outlined above. When a word is given, it is the process of determining which equivalence class it should belong to. The second step is to correlate documents with words in equivalence classes. For example, the words 'box', 'boxes', 'boxed', 'boxing', 'boxer' and 'boxers' can be considered as one equivalence class. These six words are grouped as a class in the first step. In the second step, it is decided whether to relate the documents having one of these words to the others. If a document has the word 'boxer', should this document be returned for the queries with the words 'box', 'boxes', 'boxed', 'boxing', or 'boxers'? This is not a very easy question and it has significant effect on recall and precision. Our research investigates how Google handles this second step.

Our study was conducted for three classes of words: singulars/plurals, combined words, and verbs with many postfixes. We first investigate the simplest form of stemming for singulars and plurals. We investigate how Google matches documents having one form of a word with queries having the other form of the same word. Next, we investigate how Google handles combined words in English. Combined words in English are very common and it is an open-ended question whether they should be related to their subwords. Thirdly, we investigate how Google handles words with multiple postfixes.

Our research should be helpful for many people who use search engines. There are many researchers who use search engines for data gathering or other purposes. This study should help them better understand the search engines and interpret the results more accurately. In addition, search engine optimization is an established business sector and web masters may benefit from the results in this study. Furthermore, search engine designers may find the results in this study helpful.

2. Search engines and stemming

As far as we know, this is the first study to investigate the stemming mechanisms of any search engine. Therefore, there is no direct previous work to compare our results to. However, many aspects of search engines have been studied in detail.

All leading search engines are commercial systems and their algorithms are highly valuable commercial secrets. However, the general architectures of search engines are well known [9, 10]. Search engines work in three stages: crawling, indexing and query matching. Each search engine has many crawlers running continually and discovering documents on the web [11]. They download all the indexable documents to their local repositories. Once the documents are downloaded, they are cleaned from HTML markup tags and indexed for speedy searching. For each searchable term, an inverted index file is maintained [12]. This file has pointers to all documents that have this term. Since a file has the pointers to all documents having that term, this makes matching of query terms to documents faster. When a search query is received by a search engine, it matches the terms in the query to the documents and returns the highest ranking ones. In addition to indexing, a ranking score is calculated for each document and this score is used to match the most relevant documents [13]. This is a highly simplified view of things and search engines use many methods to match the most relevant documents and serve the users with best performance. They employ thousands of machines in server farms around the world [14].

The most important stage for stemming is the indexing stage. In this stage, all the indexable terms in a document are determined. Then, a pointer for this document is added to each term's inverted

index file. As a result, this document is matched for any term in the document. Our results below indicate that a separate inverted index file is maintained for each conflated form of a word. For example, for the six different conflated forms of the word 'box', six separate index files are maintained. An index file is maintained for the word 'box', another is maintained for the word 'boxes', etc. When a document contains the word 'box', a pointer is added for this document to the index file of the word 'box'. In addition, a pointer for the same document can be added to the index files of the other conflated forms of the word 'box'. This process allows a document to be matched for queries with conflated terms. The decision to index a document for conflated forms may involve many factors and we investigate some of the possible factors in upcoming sections.

Search engines are used in many projects as data gathering tools. They provide a convenient medium to collect data about the web. Many types of data are gathered using search engines. First, they are used to gather data about the linking structure of the web in many studies [15, 16]. In addition, they are used to discover various kinds of documents on the web. One study [17] used multiple search engines to discover all pages on the web for a specific topic. Another study [18] used the search engines to discover RSS feeds on the web for issue scanning. Furthermore, search engine hit count estimations are used in many studies. In one study [19], Google hit counts are used to establish mappings among musical genres. In another investigation [20], search engine hit counts were used to discover the number of documents indexed from a particular web site.

There have been many studies to better understand search engines. Some have investigated their coverage [21, 22]. Others have investigated the freshness of search engine databases [23, 24] and the accuracy and consistency of their hit counts [25, 26]. There are many other studies to better understand various aspects of search engines. Search engines are enormous in size and complex in nature, and they constantly evolve with new algorithms and technologies.

3. Methods

We conducted this study by formulating various kinds of queries and submitting them to Google using a program that works like a web browser. It sends queries and receives the same results as the users of any web browser. Our tests were conducted at the Linux machines in Community Grids Labs of Indiana University in November and December 2008 and January 2009.

When conducting this study we needed to know the equivalence classes of English words. Given a word, we needed to know all conflated forms of that word. Such a word list was constructed in [3] for more than 18,000 word roots in English with all conflated forms. We requested this list from the author and used it in our study.

We developed a method to investigate the stemming mechanisms of search engines with two steps. In the first step, the set of documents having only the selected form(s) of a word is determined. In the second step, each retrieved document URL is checked to see whether they are indexed for the other forms of the same word. For example, when investigating whether the documents having the word 'decade' is indexed for its plural form 'decades', we first determine a set of documents having only the word 'decade' and not having the word 'decades'. Then, we check whether these documents are indexed for the word 'decades'. Since none of them has the word 'decades' in them, if they are indexed for the word 'decades', then we assume that the stemming algorithm indexed those documents for the word 'decades'.

3.1. Query construction

We formulate the queries that will return documents having only some forms of a word by using the minus sign (–) in front of the word forms that should not exist in returned documents. A sample query for words with two conflated forms can be formulated as follows:

Query: Ws –Wp

Ws: Singular form of a word

Wp: Plural form of the same word

This query returns documents in the search engine database that have the singular form of a word, but do not have the plural form of the same word. If a document has both word forms, that document is not returned.

Although many words have two conflated forms, some others have more forms. When determining the set of documents having only some forms of words, all conflated forms that should not exist in returned documents must be added to the query with minus signs in front of them. For example, when we investigate the stemming mechanisms for combined words, we require that one form of a combined word exists in returned documents and other forms and subwords do not. Such a query for the combined word 'brickbat' can be formulated as follows:

Query: brickbat –brickbats –brick –bricks –bat –bats

This query requires the returned documents to have the word 'brickbat', but not to have the words 'brickbats', 'brick', 'bricks', 'bat', and 'bats'. Similarly, the tests in section six require documents to have one form of words that have many conflated forms. Similar queries are constructed for those tests.

The tests in Section 6.1 require four forms of words in returned documents without the other conflated forms. The queries in that section had four word forms separated by a blank space and remaining conflated forms with a minus sign in front of them. A sample query for the word 'define' can be constructed as follows:

Query: defines defined defining definition –define –definitions –definer...

This query requires the word forms ('defines', 'defined', 'defining', 'definition') to exist in returned documents and other word forms not to exist.

3.2. URL selection

Once the queries are submitted to the search engine, two methods are used to select the returned documents. The first method is to select the top ranking documents. We retrieve the desired number of documents starting from the highest in ranking. The second method aims to retrieve lower ranking documents with some degree of diversity. We use a light randomization mechanism by adding a three digit random number to the query. A sample query for a word with two conflated forms can be formulated as follows:

Query: Ws TDRN –Wp

This query requires the returned documents to have a singular word Ws and not to have its plural form Wp. In addition, it requires the returned documents to have a three digit random number (TDRN). We usually retrieve 100 document URLs with this mechanism. With each query, we retrieve at most the top ranking 10 document URLs. Therefore, we submit this query with at least 10 different TDRNs. This light randomization process retrieves relatively lower ranking documents with some degree of diversity and without much difficulty.

3.3. URL checking

Once the set of URLs is determined, we check whether they are indexed for other conflated forms. In this step, we construct another query and submit it to the search engine. Google allows URLs to be submitted as queries. We construct a query with the URL and the conflated form. If the URL is url1 and the conflated form is Wp, we construct the following query:

Query: Wp url1

If the document shown by url1 is indexed for Wp, this search query returns the searched document in the result set. Otherwise, the result set does not include the searched URL. The searched URL is usually returned in the first position in the result set. However it can sometimes be returned in later positions. Therefore, we check the first 100 URLs in the result set for the searched URL. We have never encountered any cases in which the searched URL is returned in positions higher than 100.

Table 1
Results for randomly selected 10 words with singulars and plurals

Singular word	Plural word	Number of documents having singular words, returned for plural queries	Number of documents having plural words, returned for singular queries	The ratio for singular HCE/ plural HCE
benzene	benzenes	99	0	23.97
brickbat	brickbats	63	73	0.35
starling	starlings	100	62	4.28
fishmonger	fishmongers	54	67	40.17
parvenu	parvenus	65	70	1.74
nursery	nurseries	99	93	4.47
wellspring	wellsprings	96	92	0.57
mistress	mistresses	97	76	11.91
quoit	quoits	82	55	6.73
ruckus	ruckuses	94	0	792.21
	Average	**84.9**	**58.8**	

If we want to check whether the document at www.loc.gov/loc/brain/proclaim.html is indexed for the word 'decades', we submit the following query:

Query: decades www.loc.gov/loc/brain/proclaim.html

As of March 2009, the result set included this URL at the first position. This shows that this URL is indexed for the word 'decades'. This document has only the word 'decade' and does not have the word 'decades'.

In our tests, we observed that sometimes the search engine may not return the URL in the result set, even if only the URL is searched. Therefore, we first performed a search with the word form in the document and the URL. If that query returned the URL in the result set, we used that URL in our tests, otherwise it was not used.

4. Stemming for singulars/plurals

The most common form of stemming is performed between singular and plural forms of words. Singular and plural forms are very common and are semantically strongly correlated. The stemming mechanisms of search engines for singulars and plurals provide many important details of their stemming algorithm. Therefore, we first examined the stemming mechanism for singulars and plurals.

For the tests in this section, we randomly selected 10 word classes from the word list with only two forms: singular and plural. We manually checked them to make sure that they were singular and plural pairs. We first determined 100 documents having only singular words by using the light randomization mechanism explained in the previous section. We checked all documents to see whether they were indexed for plurals. Secondly, we determined 100 documents having only plurals and checked whether they were indexed for singulars.

Table 1 shows the results for singulars and plurals. The third row of the table shows the number of documents having singular words that were returned for plural queries. The fourth row shows the number of documents having plural words that were returned for singular queries. The last row shows the ratio of the singular word hit count estimations to the plural word hit count estimations. Hit count estimation is the estimated number of documents matching the searched word reported in search engine result page.

The most important observation about the results is that they do not display a simple unified pattern. The Google search engine does not return all the documents having singular words for queries with plurals or it does not return all the documents having plurals for queries with singulars.

Table 2
Results for some words with higher returning plurals

Singular word	Plural word	Documents having singular words, returned for plural queries	Documents having plural words, returned for singular queries	The ratio for singular HCE/ plural HCE
bauble	baubles	43	85	0.47
causerie	causeries	29	52	0.45
crouton	croutons	42	98	0.23
dreg	dregs	0	50	0.25

While it returns some documents having singulars for queries with plurals, it does not return some other documents having the same singulars for queries with plurals. The majority of the words display a pattern in which most of the documents are returned for queries with the other form. On average, 85% of documents having singulars are returned for queries with plurals and 59% of documents having plurals are returned for queries with singulars.

Two of the words display a different pattern for documents having plural forms. None of the documents having the words 'benzenes' and 'ruckuses' is returned for queries with their singular forms. These two words have very high singular to plural hit count estimation (HCE) ratios as can be seen in the last column of Table 1. This shows that the singular forms of these words are much more frequently used than their plural forms. This might be a factor for the different results. However, this alone cannot explain the whole reason since there is another word 'fishmonger' with a very high singular to plural HCE ratio and it displays a different pattern. Most of the documents having the word 'fishmongers' are returned for queries with its singular form.

These results show that the documents having singulars are returned more often than the documents having plurals. This motivated us to look for some words for which more of the documents having plurals are retuned for queries with singulars. We wanted to know whether there are any words for which more of the documents having plurals are returned. We selected some words with plural forms more common than singular forms. We determined four such words (Table 2). These results demonstrate that the documents having singulars are not always returned more.

The most important result of the tests in this section is that Google makes a decision for each document separately whether to return it for the conflated forms of the words in it. While it returns some of the documents for the conflated form of a word, it does not return some others. For example, while a document having the word 'nursery' may be returned for the word 'nurseries', another document having the word 'nursery' may not be returned for its plural. This decision can be made either when a search is being performed or when the document is being indexed. Our results indicate that this decision is made at indexing, and not at searching, time. If the decision had been made at searching time, the result could have been dependent on the type of queries being submitted. However, our tests show that the result is independent of the queries being submitted. As long as the queries match the document and it is high in ranking, it is always returned in the result set. For example, the document at the URL ircarchive.info/ubuntu/2007/4/21/296.html has the word 'crouton' and does not have the word 'croutons'. If we submit the following query:

Query: crouton ircarchive.info/ubuntu/2007/4/21/296.html

this document is returned in the result set. However, if we submit the query:

Query: croutons ircarchive.info/ubuntu/2007/4/21/296.html

this document is not returned in the result set. It seems that this document is indexed for the singular form of the word but is not indexed for the plural form of the same word. We can submit another query that targets this document to see whether it is still not returned for the plural form. We can construct this query by selecting some words in that document. The following query returns the URL in the result set:

Query: crouton eztk fabbos

We can send the same query with the plural form of the same word as follows:

Query: croutons eztk fabbos

This query does not return the document. Many other queries can be formulated to target this document and each one yields the same result. This shows that the decision of not returning this document for the plural form of the word is not dependent on the type of query being submitted. This decision is most probably made when indexing the document. This document is probably not indexed for the plural form of this word and it is never returned for queries with the plural form. This also makes sense for performance reasons since the decision to index a document for the conflated forms of words may involve many parameters, such as the ranking of the document, the number of occurrences of the word in the document, occurrences of other conflated forms of the word in the document, the place of the word in the document, proper name and abbreviation detection and combined word processing. In addition to the terms in documents, search engines may also use the terms in anchor texts of the links pointing to a document when making a decision whether to index for conflated forms. Moreover, Google collects user behaviour data about which links are clicked in result pages and it may use these data when deciding to index for conflated forms. We will investigate some of these factors below.

These results also indicate that Google should have a separate index file for each conflated form of a word if it is using inverted index files internally, as suggested by many publications [9, 12]. It cannot have a single index file for all conflated forms of a word as the same set of documents would be returned for each conflated form. This is not the case since there are many documents that are returned for a word, but they are not returned for the conflated forms. Many stemming papers [27] point out that stemming reduces the size of index files considerably. They propose to use one index file to hold pointers to all documents having any of the conflated forms of a word. However, this process results in information losses since it is not known which document has which form. Since Google seems to maintain separate indexes for each conflated form, the size of indexes should actually increase with stemming. Some documents should be indexed for both the words they have and their conflated forms.

4.1. The impact of document ranking on stemming

We now investigate whether higher ranking documents are more likely to be indexed for conflated forms. For this test, we randomly selected another 40 pairs of words, in addition to the 10 word pairs used in the previous section with singular and plural forms. We used these 50 word pairs to conduct the tests. In this case, we first retrieved top ranking 20 URLs for each word. Then, we checked each URL to see whether they are returned for queries with their conflated forms. In the second step, we selected 100 lower ranking documents using the light randomization mechanism. Table 3 shows the percentages of returned documents for 50 words.

On average, 97.3% of the top ranking documents having singular words are returned for queries with plurals. However, 87.3% of lower ranking documents having singulars are returned for queries with plurals. The return percentage of lower ranking documents is 10% less than the return percentage of top ranking documents. A similar pattern is observed for documents having plurals. In that case, the difference is 14%. Therefore, these results show that the Google stemming algorithm is most probably using the ranking of a document as a factor when deciding whether to index it for conflated forms. Higher ranking documents seem to be more likely to be indexed for conflated forms.

One may wonder whether Google uses the ranking as the only deciding factor when stemming. This is very easy to check. We examined the test results and saw many cases in which higher ranking documents are not indexed but lower ranking documents are. This shows that Google does not solely depend on the ranking of a document when deciding to index a document for conflated forms.

4.2. Relating singulars and plurals

The tests in previous sections imply that the documents having singular words are more likely to be indexed for their conflated forms than the documents having plural words. However those results

Table 3
Comparison of top and lower ranking documents

Documents having singulars returned for plurals		Documents having plurals returned for singulars	
Top URLs	Lower ranking URLs	Top URLs	Lower ranking URLs
97.3%	87.3%	67.6%	53.5%

Table 4
Stemming results for words with similar HCEs

Documents having singulars returned for plurals	Documents having plurals returned for singulars
86.9%	56.8%

were for randomly selected words and for those words singular forms were more common than plurals. Therefore, we determined 50 words with singular and plural forms with very similar hit count estimates. We retrieved 100 documents with the light randomization mechanism for each word and determined the number of returned documents for the conflated forms. The average numbers of indexed documents for 50 words are shown at Table 4. While 86.9% of the documents having singulars are indexed for plurals, only 56.8% of the documents having plurals are indexed for singulars. These results indicate that the Google stemming algorithm is more likely to index documents having singulars for their plurals and less likely to index documents having plurals for their singulars.

4.3. Stability of results overtime

In this section, we present the results of repeated tests for the previous two sections. We would like to briefly discuss the stability of the Google stemming algorithms over time. We performed the tests in Sections 4.1 and 4.2 a second time in March 2009, three months after the initial tests. We used the same sets of 50 word pairs with light randomization mechanism. We only repeated the tests for lower ranking URLs for the randomly selected 50 word pairs in Section 4.1.

Table 5 shows the results for randomly selected word pairs. When we compare the results of Tables 3 and 5, we observe a slight decrease in document return percentages. The result for documents having singulars has decreased from 87.3% to 84.8%, and the result for documents having plurals has decreased from 53.5% to 50.3%. A similar pattern also exists in the results for words with similar HCEs in Tables 4 and 6. The result for documents having singulars has decreased from 86.9% to 82.2%, and the result for documents having plurals has decreased from 56.8% to 55.0%.

In all four cases, document return ratios have decreased by a few percent in this time period. The decrease may either be the result of Google's slight modification of its stemming algorithms or the result of changes in the properties of crawled documents. In any case, these results suggest that the Google stemming algorithms have not changed fundamentally in this time period.

5. Stemming for combined words

Combined words are very common in the English language. They are formed of two or more subwords. For example, the word 'airway' is the combination of two subwords, 'air' and 'way'. Documents having only combined words may be returned for queries with subwords and documents having only subwords may be returned for queries with combined words. We now investigate how Google handles combined words.

We randomly selected 10 combined words that have only singular and plural forms. We then searched for documents having only singular or plural combined words. We retrieved the top 100

Table 5
Stemming results for repeated tests of 50 randomly selected word pairs

Documents having singulars returned for plurals	Documents having plurals returned for singulars
84.8%	50.3%

Table 6
Stemming results for repeated tests of 50 word pairs with similar HCEs

Documents having singulars returned for plurals	Documents having plurals returned for singulars
82.2%	55.0%

document URLs. The search queries required the combined word to exist in returned documents and for all subwords and conflated forms not to exist.

After retrieving URLs for both singular and plural forms of the combined words, each retrieved URL was checked for all combinations of subwords. If url1 is a URL retrieved for the query 'brick-bat', we submitted the following four queries to the search engine:

Query 1: brick bat url1

Query 2: brick bats url1

Query 3: bricks bat url1

Query 4: bricks bats url1

Table 7 shows the results for 10 combined words. The first five columns of the table show the results for documents having only singular combined words and the last five columns show the results for the documents having only the plural combined words. Sw1 denotes the first subword and sw2 denotes the second. Sw1s and sw2s denote the plural forms of the first and second subwords.

The results show that the documents having singular combined words are mostly returned for queries with singular subwords. For example, the majority of the documents having the word 'brick-bat' were returned for queries with 'brick bat', but none of them are returned for queries with 'brick bats', 'bricks bat' or 'bricks bats'. On average, 94.9% of documents having singular combined words are returned for queries with singular subwords. However, almost no documents are returned for queries with other singular and plural subword combinations.

The documents having plural combined words are mostly returned for queries with singular first subwords and plural second subwords. These documents are not returned for queries with other subword combinations with the exception of documents having the keyword 'airways'. The documents having this keyword are returned for queries having 'air way' and 'air ways'.

In general, these tests indicate that the Google stemming algorithm tends to index documents having combined words for their direct subwords. However, it can also occasionally index the documents for conflated forms of the subwords.

In the second round of tests, we retrieved documents having only two forms of subwords and checked whether they were returned for queries with combined words.

Table 8 shows the results. The first three columns show the results for documents having only singular subwords. The second set of three columns show the results for documents having only singular first subwords and plural second subwords. The third set of three columns show the results for documents having plural first subwords and singular second subwords. The last set of three columns show the results for documents having plural first and second subwords. Similar to the tests above, top 100 URLs are retrieved for testing.

Table 7
Results for subword queries for documents having combined words

	sw1 sw2	sw1 sw2s	sw1s sw2	sw1s sw2s		sw1 sw2	sw1 sw2s	sw1s sw2	sw1s sw2s
brickbat	93	0	0	0	brickbats	0	24	0	0
starling	61	0	0	0	starlings	0	31	0	0
fishmonger	98	0	0	0	fishmongers	0	97	0	0
wellspring	100	0	0	0	wellsprings	0	92	0	0
penthouse	100	1	3	0	penthouses	0	98	0	0
airway	99	1	0	0	airways	99	100	0	3
inflow	100	0	0	0	inflows	0	88	0	0
gumdrop	98	0	0	0	gumdrops	0	100	0	0
beachhead	100	0	0	0	beachheads	0	48	0	0
oddball	100	0	0	0	oddballs	0	100	0	0
Average	**94.9**	**0.2**	**0.3**	**0**		**9.9**	**77.8**	**0**	**0.3**

The general rule outlined above seems to apply for documents having only subwords. The documents are mostly returned for queries with direct combinations of subwords. However, more of the documents having two singular subwords are returned for the plural forms of the combined words. In addition, more of the documents having singular first subwords and plural second subwords are returned for queries with singular combined words. Google seems to index documents having two subwords for their direct combinations and also for the conflated forms of their combined words.

6. Stemming for verbs with multiple postfixes

In this section, we investigate the stemming mechanism for verbs with multiple postfixes. Verbs in English have many forms with various postfixes. We investigate the stemming mechanisms for 12 conflated forms of regular verbs. Each conflated form of a verb is generated by a postfix. These verb forms and the postfixes are shown in Table 9. We selected some of the most commonly used verb forms. We selected the verb forms for four tenses. In addition, we selected commonly used postfixes for noun, adjective and adverb generations. Although English language has much more postfixes, these ones should provide us enough information to understand the general mechanisms of Google stemming algorithm.

We determined 10 different verbs that have all 12 forms. These verbs were: define, derive, describe, elect, acquire, appreciate, distribute, evoke, imitate and manipulate. For each form of verbs, we retrieved the top 20 document URLs that have only one form of these verbs and do not have any of the other forms. For example, we retrieved 20 URLs that have only the word 'define', but do not have any other conflated form of it. Similarly, we retrieved 20 URLs for each one of the 12 conflated forms of the word 'define'. In total, 240 URLs were retrieved for 12 different forms of the word 'define'. Overall, 2400 URLs were retrieved for all 10 verbs.

In the second step, all retrieved URLs were checked to see whether they were indexed for the other conflated forms of the same verb. If url1 shows a document that is returned for the verb 'define', we submitted the following queries to the search engine:

Query 1: defines url1

Query 2: defined url1

Query 3: defining url1

…

Query 11: definitiveness url1

All URLs were checked in this manner for all conflated forms. Since each URL was checked for 11 different conflated forms, in total 26,400 (11*2400) search requests were submitted to the search engine.

Table 8
Results for combined word queries for documents having only subwords

sw1 sw2	Singular combined	Plural combined	sw1 sw2s	Singular combined	Plural combined	sw1s sw2	Singular combined	Plural combined	sw1s sw2s	Singular combined	Plural combined
brick bat	70	70	brick bats	52	52	bricks bat	0	0	bricks bats	0	0
star ling	2	2	star lings	1	1	stars ling	0	0	stars lings	1	0
fish monger	59	61	fish mongers	99	99	fishes monger	0	0	fishes mongers	0	0
well spring	22	22	well springs	1	33	wells spring	0	0	wells springs	0	0
pent house	78	0	pent houses	1	50	pents house	3	3	pents houses	0	0
air way	77	8	air ways	4	89	airs way	0	0	airs ways	0	0
in flow	83	0	in flows	0	0	ins flow	0	0	ins flows	0	0
gum drop	100	100	gum drops	1	86	gums drop	1	1	gums drops	0	0
beach head	97	97	beach heads	0	16	beaches head	2	2	beaches heads	2	2
odd ball	97	97	odd balls	0	67	odds ball	1	1	odds balls	1	1
Average	**68.5**	**45.7**		**15.9**	**49.3**		**0.7**	**0.7**		**0.4**	**0.3**

Table 9
Twelve verb suffixes used for stemming investigations

Postfix	Explanation	Sample word
-	Plain form without any postfixes	define
-s	Present tense form	defines
-ed	Past tense form	defined
-ing	Continuous tense form	defining
-tion/-sion	Noun form	definition
-tions/-sions	Plural form of the noun form	definitions
-or/-er	Doer form for actions	definer
-ors/-ers	Plural form of the doer form	definers
-able	Adjective form	definable
-tive/-sive	Adjective form	definitive
-ly	Adverb form from adjective form	definitively
-ness	Noun form from adjective form	definitiveness

Table 10 shows the average return percentages for 10 verbs. Each row, except the first one, shows the results for documents having one form of all verbs. For example, the second row shows the results for documents having the plain form of the verbs and the third row shows the results for documents having the present tense form of the verbs. The first column shows the form of verbs that occur in documents. Each column, except the first one, shows the results for queries having one form of the verbs. For example, the second column shows the percentage of documents that are returned for queries having the plain forms of the verbs and the third column shows the percentage of documents that are returned for queries having the present tense forms of the verbs. The first row shows the form of verbs that are used in queries to check documents. The last column shows the average of all cells in each row and the last row shows the average of all cells in each column.

There are many important results in this table. First, many cells have very small values. Among 132 cells in total, 19 have the value of zero and 72 have values less than 10%. There are only 16 cells with the value of more than 50%. These results show that the majority of the verb forms are very loosely correlated. Google tends to be very conservative when indexing documents for conflated forms. It seems that Google indexes documents having a conflated form of a verb for only other conflated forms that are semantically strongly related.

The cells in the grey area have much higher values than the cells in the other parts of Table 10 in general. This shows that these five verb forms are strongly correlated. These are plain form, present tense form, past tense form, continuous tense form and noun form with the postfixes -tion/-sion. The documents having one of these five forms are more likely to be indexed for the other four forms. Other higher value cells are the intersections of singulars and plurals. The postfixes -tion/-tions and -or/-ors are strongly correlated. This should be expected since singular and plurals are strongly correlated. There are two more cells with higher values. These cells show that the documents having the adjective form of the verb with -tive postfixes are highly likely to be indexed for verb forms with -ly and -ness postfixes.

The last column shows the average values of each row. It shows that the documents having four tense forms and the noun form with -tion/-sion postfixes are indexed the most for other forms of verbs. On the other hand, the documents having the verb forms with postfixes -able, -ly and -ness are indexed the least for other verb forms. The last row shows the average values of each column. It shows that the queries with four verb forms and the noun form with -tion/-sion postfix return the highest number of documents. However, the queries with the verb forms with postfixes -able and -tive return the least number of documents.

6.1. The impact of the occurrences of multiple conflated forms on stemming

As part of the tests for verbs, we also performed some tests to investigate the effects of the occurrences of multiple forms of a verb in documents. In particular, we investigated whether the occurrence of multiple forms of a verb in documents increases their likelihood of being indexed for

Table 10
Stemming test results in percentages for 10 verbs with 12 different postfixes

	Plain	-s	-ed	-ing	-tion	-tions	-or	-ors	-able	-tive	-ly	-ness	Average
Plain		99.5	62	88.5	42	3.5	31	22	13.5	4	4.5	4.5	34.1
-s	59.5		40.5	69	20	1	0	0.5	0	0	0	0	17.3
-ed	74.5	99		35.5	30	15	5.5	13.5	3.5	23.5	4	3	27.9
-ing	51	60.5	38.5		38	1	0.5	0.5	0.5	0.5	0.5	0.5	17.5
-tion	72	22.5	41.5	49		100	32.5	12	5	13	5.5	6	32.6
-tions	2.5	3	22	3	58		3	2.5	1	2.5	2.5	1.5	9.2
-or	3	10.5	9.5	1	20	1		99	0.5	1	1	1	13.4
-ors	1	1	11.5	11	21	2.5	99.5		1.5	1.5	1.5	1.5	14.0
-able	2.5	2	10.5	1	21	1.5	0	0		0.5	0	0	3.6
-tive	7.5	6	11.5	5.5	19	6.5	5.5	4	3		89	80	21.6
-ly	1	0.5	10.5	0	18.5	0	0	0	0	0.5		0	2.8
-ness	2	0.5	8.5	0.5	17	1	0	0	0	0.5	0		2.7
Average	**25.1**	**27.7**	**24.2**	**24**	**27.7**	**12.1**	**16.1**	**14**	**2.6**	**4.3**	**9.9**	**8.9**	

another form. We searched for documents that have four forms of a verb and do not have the fifth form. Then, we checked whether those documents are returned for the fifth form. We compared the results in this section with the results in the previous section.

To get the documents that have four forms of a verb ('defines', 'defined', 'defining', 'definition'), but do not have the fifth one ('define'), a sample query can be constructed as follows:

Query: defines defined defining definition –define –definitions –definer …

Similar queries were constructed to obtain documents that have other four forms. However, the documents that were returned for these types of queries did not always include all four forms of the verbs. Some documents had all four forms, some had only three forms and some had only two or one form. When multiple forms of a word exist in a query, Google seems to apply stemming and return documents that are indexed for all words in the query. Those documents may not have all the terms but they need to be indexed for all terms. On the other hand, Google is strict on exclusion. If a query has a term with a minus sign in front of it, no returned documents seem to have that term. Therefore, the documents that are returned in this section had at least one form and more commonly two or three forms.

Table 11 shows the average results for 10 verbs. Since the documents are returned for four forms of the verbs, they are checked only for the fifth form. Each row shows the results for the documents that are returned for four verb forms. For example, the second row shows the result for documents that are returned for queries with four verb forms with the postfixes of -s, -ed, -ing and -tion.

If the occurrence of multiple forms of a verb increases the likelihood of a document being indexed for another form, the value at each column of Table 9 should be higher than all the values of the same column of the grey area of Table 10. The results in Table 10 are for documents having only one form of the verb and the results in Table 11 are for documents having at least one form of the verb. The value at the second column of Table 11 is 89 and it is higher than all the values at the second column of Table 10. Similarly, the values at each column of Table 11 are either equal to or higher than the values at the corresponding columns of Table 10. These results indicate that the occurrences of multiple forms of verbs increase the likelihood of documents being indexed for other forms.

7. Conclusions and discussions

In this study we investigated the stemming mechanisms of the Google search engine. Our results indicate that Google uses a document-based algorithm for stemming. It evaluates each document separately and makes a decision whether to index or not for the conflated forms of the words it has. While a document having the singular form of a word can be indexed for its plural form, another

Table 11
Results for documents returned for four conflated forms of words

Verb forms in queries	Plain	-s	-ed	-ing	-tion
s ed ing tion	89				
Plain ed ing tion		99.5			
Plain s ing tion			67.5		
Plain s ed tion				91.5	
Plain s ed ing					47.5

document having the same singular word may not be indexed for its plural form. In addition, our results indicate that this decision is most probably made at indexing, rather than searching, time. Once it is decided that a document is to be indexed for a conflated form, it is always matched for queries having that term independent of the query being submitted.

Google may be evaluating many parameters when deciding to index a document for conflated forms of words. We investigated the impact of ranking for stemming and discovered that higher ranking documents are more likely to be indexed for conflated forms. In addition, we investigated the impact of the occurrences of multiple forms of words in documents and established that the documents having multiple forms of words are more likely to be indexed for conflated forms. Google may be using these and other factors when deciding to index a document for conflated forms.

Our results also indicate that the indexing likelihood of the Google stemming algorithm for two forms of words can be different. The indexing rate of documents having singular words for plurals is more than the indexing rate of the documents having plurals for singulars. Google may be assuming that the users who are searching for plural words might be more interested in documents having singulars. On the other hand, the users who are searching for singular words may be less interested in documents having plurals. Or they may have an algorithm that produces this result automatically by processing the documents and search queries from users. The same pattern holds for other conflated forms. For example, the documents having words with the -tive postfix are indexed frequently for words with the -tively postfix. However, the opposite is not true.

Combined word tests showed that the Google stemming algorithm tends to index the documents having combined words for their direct subwords. It is unlikely to index the documents for the conflated forms of the subwords. Similarly, the documents having subwords are more likely to be indexed for their direct combinations. The tests for verbs showed that Google tends to be very conservative when indexing documents for conflated forms. It seems to index documents having a conflated form of a verb for only other conflated forms that are semantically strongly correlated. The documents having one of the five forms of verbs are indexed the most for other forms. These are plain form, present tense form, past tense form, continuous tense form and noun form with the postfix of -tion/-sion.

The aim of this study is to help users and researchers better understand the stemming mechanisms of the Google search engine and consequently make better use of its services and interpret its results more accurately. We do not try to reverse engineer the details of Google stemming algorithms. We only try to understand the general principles behind its stemming mechanisms.

Further research can be conducted to examine other features of Google stemming algorithms and the stemming mechanisms of other search engines. In particular the effect of stemming in recall can be investigated. Our results in Section 4.1 show that 97.3% of the top ranking documents having singulars are indexed for plurals. However, we do not know in general what percentage of the documents having singulars is indexed for plurals. This may have shown the increase in recall for plurals. The increase in recall can be even greater for words with many conflated forms. There can be two methods to investigate this and neither of them is trivial. All the documents having various conflated forms of words can be retrieved from the search engine database and they can be analysed to determine the increase in recall. However, previous research has shown that it is not really possible to retrieve all documents having one term [28] from a search engine database. Another possibility is to retrieve random documents having conflated forms of words. By analysing these documents

the increase in recall can be estimated. However, random document retrieval from search engine databases is also a challenging task and requires significant effort [29].

In the beginning one of the main advantages of Google was its discovery of the implicit knowledge in the linking structure of the web, and the development of the PageRank algorithm [9]. Another was the discovery of the hidden knowledge in anchor texts of web links and using them to associate pointed documents with search queries. Later studies [30] have shown that anchor texts are similar to the document titles given by other people. Today Google and other search engines continue to use anchor texts for document discovery. Google should also be using anchor texts when deciding to index documents for conflated forms. However, we do not know the details of its algorithm for the handling of anchor texts and further research is needed to better understand the effects of anchor texts on stemming.

General purpose search engines provide search services not only in English but also in many other languages. Each language has its own challenges for search and stemming. English and many other European languages are inflectional. New word forms are constructed by adding fairly uniform suffixes. However, the degree of inflections changes for each language. English is one of the least inflectional European languages and Uralic languages, for example Finnish, are more inflectional. Arabic and Hebrew languages are also highly inflectional [31]. In addition to inflections, diacritics are also important for stemming. Although English does not have diacritics, with the exception of the apostrophe, many other languages use them frequently. Two previous studies [31, 32] have shown that while global search engines including Google usually omitted diacritics, local search engines tended to handle them better. As of March 2009, we are not aware of any published work or online resource showing that Google provides stemming for other languages. However, they may already be using stemming for some languages and they may also introduce stemming for other languages in upcoming years. We believe that this study would also be very helpful to understand the Google stemming mechanisms for other languages.

Acknowledgements

I would like to thank the director of Community Grids Labs at Indiana University, Dr Geoffrey C. Fox, for allowing me to use the Linux cluster in his lab to perform the tests on this study. I would also like to thank Mark Kantrowitz for graciously sharing with me the manually generated English word list with equivalence classes.

References

[1] J.B. Lovins, Development of a stemming algorithm, *Mechanical Translation and Computational Linguistics* 11(1) (1968) 22–31.

[2] M. Porter, An algorithm for suffix stripping, *Program* 14(3) (1980) 130–137.

[3] M. Kantrowitz, B. Mohit and V. Mittal, Stemming and its effects on TFIDF ranking, *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Athens, Greece, 2000) 357–359.

[4] M. Hafer and S. Weiss, Word segmentation by letter succession varieties, *Information Storage and Retrieval* 10 (1974) 371–385.

[5] M.F. Porter, *Snowball: A Language for Stemming Algorithms*. Available at: http://snowball.tartarus.org/texts/introduction.html (accessed January 2009).

[6] *The Essentials of Google Search*. Available at: www.google.com/support/websearch/bin/static.py?page=searchguides.html&ctx=basics (accessed January 2009).

[7] *The Official Blog of the Live Search Team at Microsoft, 'Do what I mean, not what I say!'*. Available at: http://blogs.msdn.com/livesearch/archive/2007/10/24/do-what-i-mean-not-what-i-say-part-1-of-2.aspx (accessed January 2009).

[8] *Review of Yahoo! Search*. Available at: www.searchengineshowdown.com/features/yahoo/review.html (accessed January 2009).

[9] S. Brin and L. Page, The anatomy of a large-scale hypertextual web search engine, *Computer Networks and ISDN Systems* 30(1–7) (1998) 107–117.
[10] A. Arasu, J. Cho, H. Garcia-Molina, A. Paepcke and S. Raghavan, Searching the web, *ACM Transactions on Internet Technology* 1(1) (2001) 2–43.
[11] G. Pant, P. Srinivasan and F. Menczer, Crawling the web. In: M. Levene and A. Poulovassilis (eds), *Web Dynamics: Adapting to Change in Content, Size, Topology and Use* (Springer-Verlag, Heidelberg, 2004).
[12] J. Zobel and A. Moffat, Inverted files for text search engines, *ACM Computing Surveys* 38(2) (2006) 1–56.
[13] A.N. Langville and C.D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings* (Princeton University Press, Princeton, NJ, 2006).
[14] L.A. Barroso, J. Dean and U. Holzle, Web search for a planet: the Google cluster architecture, *IEEE Micro* 23(2) 2003 22–28.
[15] L. Vaughan and M. Thelwall, Scholarly use of the web: what are the key inducers of links to journal web sites?, *Journal of the American Society for Information Science and Technology* 54(1) (2003) 29–38.
[16] I.F. Aguillo, B. Granadino, J.L. Ortega and J.A. Prieto, Scientific research activity and communication measured with cybermetrics indicators, *Journal of the American Society for Information Science and Technology* 57(10) (2006) 1296–1302.
[17] J. Bar-Ilan, B.C. Peritz, Evolution, continuity, and disappearance of documents on a specific topic on the web: a longitudinal study of 'informetrics', *Journal of the American Society for Information Science and Technology* 55(11) (2004) 980–990.
[18] M. Thelwall, R. Prabowo and R. Fairclough, Are raw RSS feeds suitable for broad issue scanning? A science concern case study, *Journal of the American Society for Information Science and Technology* 57(12) (2006) 1644–1654.
[19] R. Gligorov, Z. Aleksovski, W. ten Kate and F. van Harmelen, Using Google distance to weight approximate ontology matches, *Proceedings of the 16th International Conference on the World Wide Web 2007* (Banff, Alberta, Canada, 2007).
[20] L. Vaughan and M. Thelwall, Search engine coverage bias: evidence and possible causes, *Information Processing and Management* 40(4) (2004) 693–707.
[21] S. Lawrence and C.L. Giles, Accessibility of information on the web, *Nature* 400 (1999) 107–109.
[22] A. Gulli and A. Signorini, The indexable web is more than 11.5 billion pages, *Proceedings of the 14th International Conference on the World Wide Web 2005* (China, Japan, 2005).
[23] D. Lewandowski, H. Wahlig and G. Meyer-Bautor, The freshness of web search engine databases, *Journal of Information Science* 32(2) (2006) 131–148.
[24] D. Lewandowski, A three-year study on the freshness of web search engine databases, *Journal of Information Science* 34(6) (2008) 131–148.
[25] M. Thelwall, Quantitative comparisons of search engine results, *Journal of the American Society for Information Science and Technology* 59(11) (2008) 1702–1710.
[26] A. Uyar, Investigation of the Accuracy of Search Engine Hit Counts, *Journal of Information Science*, DOI: 10.1177/0615551509103598, Prepublished April 24, 2009.
[27] B.L. Narayan and S.K. Pal, Distribution based stemmer refinement, *Lecture Notes in Computer Science* 3776 (2005).
[28] M. Thelwall, Extracting accurate and complete results from search engines: case study Windows Live, *Journal of the American Society for Information Science and Technology* 59(1) (2008) 38–50.
[29] Z. Bar-Yossef and M. Gurevich, Random sampling from a search engine's index, *Journal of the ACM* 55(5) (2008) 1–74.
[30] N. Eiron and K.S. McCurley, Analysis of anchor text for web search, *Proceedings of 26th ACM SIGIR* (2003) 459–460.
[31] J. Bar-Ilan and T. Gutman, How do search engines respond to some non-English queries? *Journal of Information Science* 31(1) (2005) 13–28.
[32] E. Toth, Exploring the capabilities of English and Hungarian search engines for various queries, *Libri* 56(1) (2006) 38–47.