A Method for Extracting Word Lists from the 361-billion Token Google Books Corpus

Malcolm Prentice*

Abstract

The recent release by Michel et al. (2011) of n-gram data on the Google Books corpus allows the creation of new English word frequency lists based on one of the largest collections of text in the world. However, the amount of data involved creates a number of technical problems. This article describes the process of creating a software tool to overcome those problems, and then provides instructions for extracting word lists of various kinds. The tool is free to download, adapt and share under an open-source software license. An example of what can be done – an update of existing BNC and GSL/AWL word frequency lists using data from the corpus – is given at the end of the article.

Introduction: About Word lists and Corpora

Lists of the most frequent words or word families in English can be useful for learning and teaching, whether memorized directly or used indirectly to create level appropriate materials and tests. Lists used include the General Service List (West, 1953) based on the Brown corpus (Francis & Kucera, 1979), the Academic Word List supplement to the GSL (Coxhead, 2000), the British National Corpus lists (Leech Rayson & Wilson, 2001; Nation, 2006) and the JACET 8000 (Ishikawa et al., 2003), also based on the British National Corpus but supplemented with six million tokens of text targeted at the needs of Japanese students. The lists above are the ones most commonly seen in the Japanese EFL University context - while larger and more up-to-date alternatives do exist, they have restrictions on access and distribution that limit their utility in the classroom.

Each word list is based on a corpus, and inherits strengths and weaknesses from that corpus. This article describes a process for extracting a new word frequency list from the Google Books corpus using lists prepared and released by Michel et al. (2011). These lists have a number of

* 非常勤講師／英語教育

weaknesses, but also the unique strength of being drawn from a corpus several thousand times larger than those mentioned above.

The Google Books corpus and released data

According to Michel et al. (2011), 15 million written works from the collections of 40 university libraries and a number of publishers have been scanned and processed using Optical Character Recognition software, as part of a Google project (Google Books). Of these, Michel et al. (2011) chose five million high quality scans with good metadata, representing around 4% of the books ever published. In the resulting corpus, there are 361 billion English tokens from 3,288,288 books written between 1520 and 2008. The aim of Michel et al. (2011) is the study of cultural trends by following the changes in language use that result from censorship, shifts in diet, flu pandemics and so on. In linguistics they have used it to track the process of verb regularization and quantify how well dictionaries have represented actual word usage at the time of their publication.

Given its size and the copyright status of much of its content, the corpus itself cannot be distributed. What has been released are one to five word chunks, or *n-grams*: unigrams, bigrams, trigrams, 4-grams and 5-grams, listed by year with counts. N-grams occurring less than 40 times in the corpus have been excluded, to avoid including an endless list of low frequency scanning errors, foreign words and so on.

The unigram data looks like this:

circumvallate	*1978*	*313*	*215*	*85*
circumvallate	*1979*	*183*	*147*	*77*

The 5-gram data looks like this:

‘ t at all common	*2007*	*33*	*33*	*25*

As explained on the data download page (http://ngrams.googlelabs.com/datasets), this tells us that the word *circumvallate* occurred 313 times on 215 pages of 85 books in 1978, and 183 times on 147 pages of 77 books in 1979. The 5-gram example occurred 33 times on 33 pages of 25 books in 2007 - note that the definition of *token* includes punctuation and ‘t’.

Four sub-lists of the full data set (*eng-all*) are available: lists limited to books published in the US (*eng-us*), limited to books published in Britain (*eng-gb*), limited to fiction (*eng-fiction*), and the “Google Million” (*eng-1m*) limited to a random selection of 6000 good quality scanned books

from each year.

The Google Books Corpus n-gram data is not a corpus and cannot replace one. Unlike the corpora mentioned above, the tokens are not tagged with part-of-speech information. Concordancing (see http://googlebooks.byu.edu/ for a prototype) can offer at most two words of context either side of the target word. The most important issue for the purpose of this article, however, is that the sample of language is unknown.

We do know that 27% of tokens in the *eng-all* list come from just the last eight years, and that a lot of books come from university libraries. Then again, for all we know the books our students read are also 27% from 2000-2008 and largely from university libraries. The problem is that unlike the corpora mentioned above, we do not really know what is in the Google Books corpus. Agree or disagree with the choices made while collecting the other corpora, we can make informed judgements about how representative they are of the language our students will encounter. For the Google Books corpus we cannot. Despite these disadvantages, the sheer size of data sets like this makes them worth investigating (Halevy, Norvig & Pereira, 2009; Michel et al., 2011), for example as they include words smaller collections miss.

The purpose of this project was initially just to satisfy my curiosity. I wanted to take the full list including both American and British English, extract tokens and counts since 1994 (when the BNC stopped collecting), and find which of the most frequent tokens on that list were missing from lists of the top three thousand BNC word families and from the GSL/AWL lists. The GSL includes the less than useful word “shilling”, and neither the GSL, AWL nor any of the 14 BNC lists include the word “internet”. Corpora and word lists age, and I wanted an update.

Meanwhile, however, a colleague who was beta testing an early version of the script created a comprehensive list of American English tokens since 1959. Another colleague was interested in the bigram lists. Since the sub-list, n-gram type and year that each teacher might be interested in apparently varies, the aim of this article is not to give a definitive “Google Books Word List”, but to offer an open-source software tool for extracting the word list you need from the data yourself. The answer to my own question (tokens missing from the GSL/AWL and BNC lists that might be worth learning) is given as an example at the end of the article.

The sheer quantity of data is the reason a purpose-built tool is needed. The n-gram lists vary in size between ten and eight hundred text files, each around 1GB, each containing millions of

lines. Opening even one of these files in a standard text editor or spreadsheet program will result in a truncated file or a frozen system. A programming language (in this case Python) was used to write a script that processes the lines one by one rather than loading them all at once. The tool is available to download at http://code.google.com/p/google-ngram-stripper/ and is released under a GNU GPLv3.0 license, meaning it is free to download, adapt and distribute.

Each token has many entries in a file - one for each year between 1520 and 2008. For each file in a folder, for each line in the file, the script checks that the "year" column is after the earliest year of interest, then combines all the remaining lines into one. It deals with occasional glitches in the data, such as years containing non numbers (*!996*), and it keeps an eye on what files are already in the folder so work is not repeated and important files are not overwritten. It then discards any tokens that occur less than a certain number of times, as in a corpus this size even spelling mistakes can accumulate 10000 occurrences. It times the processing and provides progress updates on the screen as it works. Finally, it creates a CSV format results file of manageable size that can be opened and manipulated as a spreadsheet.

Challenges in handling Google Books ngram data

Two issues had to be addressed. The first issue was that the tokens have been split into files ignoring capitalization, with variants (the, The, THE) scattered across a number of files. This is an issue if accurate counts are wanted for a specific occurrence threshold, and for all but narrow year-range unigram queries, a threshold is advisable to keep the size of the results file under control. As an example, take the word "Blighty" - a low frequency but well-known word used by British people overseas to refer to Britain, these days usually for comic effect. It occurs in three different *all-1grams* files as "Blighty" (9938), "blighty" (1493) and "BLIGHTY" (232), for a total of 11663 occurrences since 1950. However, it was excluded from a search with threshold 10000 because of the individual counts. The solution is to save everything into a temporary file, then process that temporary file to combine the versions before applying the threshold. When this method was used to repeat the query ("unigrams occurring more than 10000 times since 1950"), over 20000 new unigrams made it into the list and even high frequency tokens increased their count - for example "the" picked up around 20000 occurrences, presumably from low frequency capitalization errors such as "tHE".

The second issue was caused by the solution to the first - the temporary file itself became quite large. The solution was to split the data alphabetically into a set of year-filtered temporary files: one for each letter from A to Z, plus a 27th file containing non-alphabetic n-grams (those starting with

numbers, punctuation and so on). This has the additional benefit that the most time-consuming part of the processing is already done should the user want to rerun the script with the same year and a different threshold. While the temporary file size is not a problem for unigrams, the next section quickly looks at the hardware requirements for handling larger data sets.

About 2,3,4 and 5-grams

This article concentrates on unigrams, but as the script also works with the larger n-gram lists a few warnings are necessary. Firstly, producing alphabetic year-filtered temporary files takes, on an average system, about one minute per file. This means that while the ten *all-1grams* files only take 10 minutes, expect the 800 *all-5grams* files to take about 13 hours.

Secondly, there is a constraint on the second stage – producing the final list. The 2,3,4 and 5-gram files are the unigrams rearranged in an increasingly large number of combinations at lower and lower frequencies. As the amount of data increases, the less it can be compressed. A 1994 year-filter reduces 9GB of *all-1grams* to an easily handled 84MB of temporary files, but only reduces 733GB of *all-5-grams* to 12GB of temporary files. Large individual temporary file sizes can make it difficult for the second stage of processing to produce a final list. Having observed memory usage during several runs, it seems that around 6 megabytes of system memory are needed for each megabyte of temporary file size, putting an upper limit on the size of files that can be processed.

The largest file so far processed is the 1.8GB alphabetized “T” file from a 1994-filtered *all-5gram* data set, which requested 9.2GB of memory on top of system demands and completed in 81 seconds on a machine with 16GB of memory installed. The same machine failed to process the 3GB non-alphabetical file (5-grams starting with punctuation marks, numbers, etc.) as it requested 18GB – more than was available. When more memory is requested than is available, computers use what is called “swap” – an area of slow hard disk memory. When too many programs are open and a computer becomes extremely slow, it is often because the computer has started using swap. The processing failed to complete, even when left to run for several days, and the 3GB non-alphabetical file had to be removed from analysis.

For reference, with the 3GB problem file removed, a query for *“5-grams from the all-5grams list occurring more than 3000 times since 1994”* produced a manageable 10MB file containing 359692 entries. Here are the top ten results of that query, which perhaps tell us more about the contents of the corpus than they do about the English language.

http : / / www	4378933
i don ' t know	4101437
at the end of the	3640779
don ' t want to	1863883
i don ' t think	1842030
in the united states .	1689480
cambridge : cambridge university press	1626704
i don ' t want	1530677
i ' m going to	1512289
in the middle of the	1468471

There are several options for avoiding the temporary file size bottleneck currently being explored – finding a less memory-hungry technique, speeding up the swap memory by using a solid-state hard disk, or simply splitting the data further (Aa, Ab, Ac). In the meantime, the practical limit for a standard system with 4GB of memory is probably a temporary file size of 500MB – enough for almost any unigram query, but insufficient for almost any 5-gram query. Note that the sub-lists (UK, GB, Million, Fiction) are smaller, and so can handle much longer year ranges and lower thresholds than the full list.

Method

Preparation

The script requires the programming language Python. If you are using Mac or Linux, this is already installed. If you are using Windows, you can download it for free (http://python.org), and install it as you would any program.

Before you start, you also need to choose a list (US, UK, million, fiction or ALL), a year range (e.g. since 1994) and a threshold. Choosing a threshold can be tricky. Some OCR errors, spelling errors and foreign words have accumulated so many occurrences that they overlap with low frequency tokens. If you set the threshold too high your final results file will be full of useless tokens and perhaps too large to open – if too low you will miss tokens you want. The appropriate threshold depends on the list (e.g. all, US or GB) and year range (e.g. since 1994). One approach is to run the program once, browse one of the smaller alphabetized files it produces and get a sense of the threshold at which tokens you want turn into those you do not. For reference, a query for *"tokens occurring at least 10000 times since 1994"* will produce 134608 tokens from the all-1grams list,

68968 from the US-1grams list and 33635 from GB-1grams list.

Procedure

1) Download the list you need from http://ngrams.googlelabs.com/datasets
2) Download the most recent version of the script (“google-ngram-stripper.py”) from http://code.google.com/p/google-ngram-stripper/
3) Unzip the list files and put them in a folder along with the script.
4) Open the script with a text editor. Change the year range and the occurrence threshold at the top of the file.
5) Open a terminal window – this will be somewhere in your program list, but it is easier just to search for “Terminal”.
6) Change directory to the folder with your files using “cd”. For example, if you are using a Mac and the files are in a folder called “googlebooks” on your desktop, type “cd Desktop” then “cd googlebooks”
7) Type “python google-ngram-stripper.py"
8) Wait around 10 minutes, depending on your system.
9) If you plan to open the output file (CSV) in a spreadsheet, be careful to select the Unicode option as you import, and use the option to delimit text using only tabs, otherwise spaces and quote marks might confuse the import.

Example Use Case: “I wonder what common unigrams are missing from the BNC, GSL and AWL?”

I used the script above to produce a list of the 3000 most frequent unigrams since 1994 from the *all-1grams* list. Using a separate script developed to grade text by level (http://code.google.com/p/malc-text-grader), I made a list of unigrams that were not in lists of the most frequent 3000 word families as described in Nation (2006) and downloadable from http://www.victoria.ac.nz/lals/staff/paul-nation.aspx.

I cleaned the list, removing names, places, internet address fragments (www com http), prefixes (anti non pro sub trans un) and a surprising number of non-English words, mostly articles (und, le, von, la, los, el, san, et, der). I removed a lot of abbreviations (cm, mm, ml, mg, c.f., ibid.,

IV, i.e., ms, pp, eds., etc., inc., jr, pH, TV), but their frequency made me think that it might be worth teaching them separately at some point. Then I reduced the BNC list to 183 lemmas, which are given in Appendix 1. Of course, these words do appear later on in the BNC – mostly on lists 4-7, but scattered up to list 12, with only "internet" offlist.

I did the same for the GSL/AWL lists, which produced 157 lemmas (Appendix 2). A number of the words missing from the BNC 1, 2 and 3 lists are covered by the GSL/AWL, and vice versa.

Is it worth learning these words explicitly? According to Nation and Newton (2009:133), after the first 2-3000 word families have been learned - by methods including memorizing word lists - the teaching focus should turn to learning and coping strategies to help students deal with the lower frequency words they meet. This is because of the declining return on investment of effort – while words on the first three BNC lists account for 89.8% of all tokens in the BNC, learning the fourth list will only add another 1.79% (Nation, 2006), so learners are better served by collecting words they are actually encountering. The words in Appendix 1 are fourth list or below.

However, the list below is not of *additional* words, but a list of words that might have been higher in the GSL or BNC lists had the corpora been compiled in 2008 and included both British and American English. However, "*might have been in*" is a long way from "*should be added to*", and to say the latter we need data on the extra coverage that learning these extra words would give. Unfortunately, that is easier said than done. While we can measure the increased coverage the words would give using the BNC corpus, judging an update to the BNC lists by using the BNC corpus itself is hardly a valid test. At the same time, while we can measure coverage using the n-gram lists, doing so runs into the sampling problem described in the introduction.

To illustrate that sampling problem, here is a quick comparison. The Google Books *all-1grams* list from years 1994-2008 contains just over 134 billion tokens, of which the top 3000 most frequent tokens (not word families or lemmas) have a cumulative coverage of 80%. A similar analysis of the BNC using the Leech, Rayson and Wilson (2001) lists shows that the top 3000 tokens in the written portion of the BNC together cover around 77%. For the BNC this number is meaningful – we know what texts are included and can judge how representative that sample is of the texts our students might read. For the Google Corpus we know nothing about the sample and so cannot make confident generalizations. Again, the results are perhaps only useful as a hint to the nature of the corpus - the greater coverage by the Google Corpus suggests a lower variety of included texts than the balanced BNC.

Conclusion and next steps

In summary, if you are using the GSL/AWL and you doubt that “shilling” is still an essential word, or if your faith in the BNC lists is shaken by the absence of “internet”, then please consider the words in the appendices. Alternatively, download the new tool (http://code.google.com/p/google-ngram-stripper/) and use it to create your own lists for your own purposes.

The next step will be to organize the tokens by word family or lemma and then to lower the hardware requirements to extend the range of queries average systems can make.

Thanks

Thanks to Charles Kelly, at Aichi Institute of Technology, for testing an early version of the script. Currently hosted on his website (among many other useful things) are a list of unigrams, bigrams, trigrams and 4-grams from the US list since 1959 (http://www.manythings.org/wordfrequency).

References

Coxhead, A. (2000). A new academic word list. *TESOL Quarterly*, 34, 213–238.

Francis, W. & Kucera, H. (1979). *Brown corpus manual (Revised)*. Retrieved from http://khnt.aksis.uib.no/icame/manuals/brown/

Halevy, A., Norvig, P., & Pereira, F. (2009). The unreasonable effectiveness of data. *Communications of the ACM, 24*(2), 8–12.

Ishikawa, S., Uemura, T., Kaneda, M., Shimizu, S., Sugimori, N., & Tono,Y. (2003). JACET8000: *JACET list of 8000 basic words*. Tokyo: JACET.

Leech, G., Rayson, P., & Wilson, A. (2001). *Word Frequencies in Written and Spoken English: based on the British National Corpus*. London: Longman.

Michel, J., Shen, Y., Aiden, A., Veres, A., Gray, M., Brockman, W., The Google Books Team, Pickett, J., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M. & Aiden, E. (2011). Quantitative Analysis of Culture Using Millions of Digitized Books. *Science, 331*, 176-182.

Nation, P. (2006). How Large a Vocabulary Is Needed For Reading and Listening? *The Canadian Modern Language Review*, *63*(1), 59-82.

Nation, P. & Newton, J. (2009). *Teaching ESL/EFL Listening and Speaking*. New York: Routledge.

West, M. (1953). *A general service list of English words*. London: Longman.

Appendix 1: Tokens missing from BNC lists 1,2 and 3 but present in a list of the most common 3000 unigrams collected since 1994 in the Google Books Corpus all-1grams list

abstract academic accompany acquire acute administer agriculture anxiety appendix architecture asset bay beneath biological carbon chronic cite classify cognitive colonial communist component compose conclude conduct congress consequence consist constitution construct contemporary contrary cooperation copyright curve database decade decline decrease democracy density derive device diagnosis digital discourse disorder diversity divine dna doctrine domain dominant dynamic empirical enterprise equation equilibrium ethics ethnic evaluation evident executive expansion federal feminist fluid former formula fundamental furthermore gay gender gene guideline hence hypothesis implementation importance induce input integrate interact interface internet intervention isolated jewish journal latin latter legislation liberty linear literary liver matrix mechanism media membrane ministry mode molecular moreover muslim mutual narrative native network notion novel objective ocean online organic output oxygen pacific parallel parameter participant particles perceived personnel perspective philosophy phrase poet poverty presence president primarily primary prior professor profile protein provision publication racial radiation radical rapid ratio rational reform regime republic resolution respectively review rural sacred scholar sector sequence software spatial specify statistics subsequent summary superior supreme syndrome task temple territory theme theology thereby transition treaty underlying universe urban valley velocity vertical virtue wealth web yield

Appendix 2: Tokens missing from GSL and AWL but present in a list of the most common 3000 unigrams collected since 1994 in the Google Books Corpus all-1grams list

abuse acid acute alcohol anti appeal architecture assets atmosphere bible biological breast budget campaign cancer carbon career cash cast catholic cell christ chronic circuit click client climate clinical cognitive column communist competitive concrete congress conservative copyright counter county crisis criticism database defense democracy democratic density diagnosis digital discourse disorder divine dna doctrine dose drug electronic emotional engaged enterprise equilibrium era essay executive feminist fiction fluid gene goods guy height hell household huge infection intellectual interface internet interview jesus jewish kids laboratory landscape latin linear liver london magazine magnetic matrix membrane mission molecular muscle museum muslim narrative nerve novel objectives online opposition oral organic organizational oxygen pacific parliament patients peak personality personnel phrase port prince professor profile protein racial radiation reference reform rural scholars senior servers session software spatial species storage superior

supreme surgery symptoms syndrome technologies television territory theology therapy thou thy tissue tone traffic treaty troops turban vast velocity versus vertical vice video vital web zone

（2011.9.26 受稿，2011.10.19 受理）