A New Computer Science Academic Word List

Sani Yantandu Uba[1], Julius Irudayasamy[1], & Carmel Antonette Hankins[2]

[1] Department of English Language & Literature, Dhofar University, Salalah, Oman

[2] Foundation Program, Dhofar University, Salalah, Oman

Correspondence: Sani Yantandu Uba, Department of English Language & Literature, Dhofar University, Salalah, Oman.

Received: December 14, 2022 Accepted: February 14, 2023 Online Published: April 7, 2023

doi:10.5430/wjel.v13n5p319 URL: https://doi.org/10.5430/wjel.v13n5p319

Abstract

This corpus-based vocabulary study aimed to develop a new computer science academic word list across ten sub-disciplines of computer science defined by Association for Computing Machinery (hereafter ACM). A corpus of Computer Science containing 2,500,990 running words was developed from 300 Computer Science Research Articles (hereafter CSRAC) as a database of this study. Drawing on and combining procedures and methods from Coxhead (2000), Gardner and Davies (2014) and other previous studies, this study developed a New Computer Science Academic Word List (hereafter NCSAWL), containing the most frequently-used computer words in computer research articles from the corpus. The NCSAWL contains 444 words, which accounts for approximately 20.33% of the coverage in the CSRAC, the NCSAWL has a much better coverage of computer English. The result of this study has numerous implications for computer science learners, English teachers, researchers, as well as material writers and course syllabus designers. For examples, English computer teachers should focus on teaching learners the most high-frequent words which have a dispersed coverage and have special meaning and use in the discipline of computer science. Teachers could also raise the awareness of learners that some words have different meanings and uses in general English. The material designers for English for academic and special purposes could incorporate the NCSAWL vocabulary into their academic reading and writing materials for computer science students. Researchers and English language teachers who are interested in expanding their computer science academic vocabulary could also use this NCSAWL.

Keywords: corpus, computer science, word list, academic vocabulary, research article

1. Introduction

The purpose of this study is to provide a New Computer Science Academic Word List (NCSAWL hereafter) which would supplement the Computer Science Word List (Minshall, 2013); and Computer Science Academic Vocabulary List (Roesler, 2020). This NCSAWL aims to help computer science students to acquire more vocabulary in their major. Scholars state that non-native English speakers find it difficult to acquire academic vocabulary (Cobb & Host, 2004; & Yang, 2015). Brezina and Gablasova (2015) argue that vocabulary learning is a complex process because learners need to acquire not only the form but also different meanings of a given word. The knowledge of vocabulary has a positive impact on language learners' writing proficiency and reading comprehension (Nation, 2001). Some scholars argued that foreign language learners must acquire appropriate vocabulary size because it is the most important component in learning a foreign language (Laufer, 1992; Nation, 2006; Schmit, Jiang & Grabe, 2011; Yang, 2015; & Bi, 2020).

Nation (2013) has classified words into three main categories on the basis of frequency levels: high-frequency words, mid-frequency words, and low-frequency words. Academic words and technical words can be found in any place along the spectrum of frequency from high to low, although it is usually within the mid-frequency range (Bi, 2020). The focus of the present study is on academic vocabulary. Many scholars argue that there is a lack of an acceptable definition of academic vocabulary (Bi, 2020; Gardner & Davies, 2014; & Yang, 2015). For example, Farrel (1990) considers academic words as words which have a high-frequency, as well as having a wide range of occurrences across academic texts but are infrequent in other genres. For Gardner and Davies (2014) academic words are words which occur more frequently in academic texts than in other genres, as well as having even distribution across disciplines. The knowledge of academic vocabulary is considered very important for both reading comprehension and academic success (Corson, 1997; Goldenberg, 2008; Nagy & Townsend, 2012; Lei & Liu, 2016; & Roesler, 2020). It is also considered as one of the key factors in teaching and learning, but learners find it very difficult to acquire it (Shaw, 1991; Thurstun & Candlin, 1998; & Lei & Liu, 2016). This difficulty has motivated many researchers to develop general academic vocabulary lists and specialized academic vocabulary lists across many disciplines.

2. Literature Review

2.1 Vocabulary Lists

Many scholars report that West's (1953) General Service List (GSL hereafter) is the most widely used in teaching and pedagogical vocabulary research (Hirst & Nation, 1992; & Brezina & Gablasova, 2015). However, despite the fact that the GSL was widely used, it has been criticized by many scholars. For example, Richards (1974) claims that the list is out of date and needs revision. Richards also stated that there are inconsistencies on the GSL. For example, words such as *elephant, monkey,* and *bear* were included on the GSL and words such as *fox* and *tiger were* excluded from the list despite the fact that both words mentioned belong to the semantic field of animals. Brown (2014) also argues that the GSL is not representative of contemporary English. The compilation processes of the GSL involve both quantitative and qualitative criteria which bring subjectivity into the final list (McEnery & Hardie, 2011). Brezina and Gablasova (2015) state that some words included on the GSL are not in contemporary general use, for instance, *footman, telegraph,* and *milk-maid*, while on the other hand, words which are in general use are excluded, such as *internet, computer,* and *television*. Minshall (2013) has pointed out that there is a lack of consensus concerning the number of words on the GSL. For example, Nation and Hwang (1995) claim that the list has 2,147-word family; and Nation (2004) in another study reported 1,986-word family. Gilner (2011) also reported different number of words (1,907 main entries) on the list. Following this, two new general service lists with improved methodologies and lemma-based forms were developed by both Brown (2014) and Brezina and Gablasova (2015). Brown's (2014) new general service list (NGSL hereafter) consists of 2,801 Lemmas and Brezina and Gablasova (2015) new-general service list (new-GSL hereafter) has 2,494 lemmas developed from over 12 billion running words across four corpora (BNC, EnTenTen 12, LOB and BE06).

One of the earliest word lists was University Word List (UWL hereafter) of 836-word family developed by Xue and Nation (1984). The most widely known academic word list (AWL hereafter) was developed by Coxhead (2000) of 570-word family, which was established from various academic texts, comprising texts across four disciplines: law, science, commerce, and art; university textbooks, and research articles. However, there has been much criticism on Coxhead's AWL. For example, Gardner and Davies (2014) claim that the AWL had methodological problems and was created on word - family-based forms instead of lemma-based forms, while the latter is more informative and user friendly. Following this, a new methodology was used to create a new academic vocabulary list (NAVL hereafter). It was developed by Gardner and Davies (2014) of 3000 lemmas from 120 million running words of written academic genres.

Minshall (2013) established a Computer Science Word List (hereafter CSWL) of 433 headwords from a corpus of 3.6 million running words, comprising journal articles and conference proceedings. The CSWL combined with GSL and AWL attained 95.11%, which meets the lexical threshold of Laufer (1990). Another computer word list study was conducted by Chen and Gang (2019) where they established a technical computer science word list (TCSWL hereafter) of a 769-word type from a corpus of 10.5 million tokens. They adopted three criteria of Coxhead (2000) range, frequency and word type in selecting words. Bi (2020) also developed a computer science vocabulary list (CSVL hereafter) of a 356-word family from a corpus of 7.5 million running words. Bi also used three criteria for establishing the CSVL: frequency, range and dispersion. The study indicated that the CSVL, combined with the students' lexical repertoire met the lexical threshold of 95.16% of Laufer's (1990) proposal that the learners must have such minimum requirement for reading comprehension. Again, this researcher used word family instead of lemma-based form. Secondly, only three criteria were used in developing the corpus. Thirdly, a specialized dictionary was not used to verify whether words have special meaning and use in the discipline of computer science.

Roesler (2020) established another computer science academic vocabulary list (CSAVL hereafter) of 904 lemmas from a corpus of computer science research articles and textbooks of 3.5 million tokens. Roesler (2020) used six criteria in establishing the CSAVL: range, discipline measure, minimum frequency, dispersion, and special meaning. One of the major weaknesses of the CSAVL is the inclusion of words which do not have special meaning and use in the discipline of computer science, such words are: *explicitly, most, such, due, respectively*, and so on. In addition, the CSAVL comprises a lot of words which have special meanings only in the discipline of Mathematics, such as *coefficient, minimum, mathematics, finite, divisible* and so on. Following this, the present study aims to establish a new computer science academic word list with improved methodology by drawing on and extending procedures from previous studies (Coxhead, 2000; Gardner & Davies, 2014; Yang, 2015; & Lei & Liu, 2016). In the process of establishing this new list, we developed the following research questions:

1. To what extent are the NAWL (Gardner & Davies, 2014) and new-GSL (Brezina & Gablasova, 2015) used in the New Computer Science Academic Word List of this study?
2. To what extent do the contents of New Computer Science Academic Word List differ from Roesler's (2020) CSAVL and, which list might better serve potential computer science users?

3. Methodology

3.1 The Development of the Corpus

As mentioned above the aim of this paper is to provide a computer academic word list which could assist computer students and others in improving their reading skills and vocabulary development. A corpus of Computer Science containing 2,500,990 running words was developed from 300 Computer Science Research Articles (hereafter CSRAC) as a database of this study. Scholars have developed different criteria in establishing different academic and specialized word lists. This depends upon the objectives of the studies. We first consulted experience researchers in our University who are Faculty members in the Department of Computer Science and have been teaching for the past ten years for the selection of relevant journals and sub-disciplines of Computer Science. Following this, we selected 10 Computer sub-disciplines defined by Association for Computing Machinery (hereafter ACM) similar to some previous studies (Minshall, 2013; Bi, 2020; & Roesler, 2020). We also selected relevant journals for the ten sub-disciplines from the most widely known online database of high quality research and high impact factor journals: web of science via this link: www.sciencedirect.com. The journals were chosen if at least a keyword from the title of the journal corresponded to a name of each sub-discipline. For example, HardwareX was one of the journals chosen for Hardware sub-discipline; and International Journal of Human-Computer Studies was also selected for sub-discipline of Human-centred Computing. We accessed all the journals on our University website. The list of the ten sub-disciplines is shown in table one below. We developed another corpus of computer science for testing the reliability of our result, corpus of computer science (hereafter, CSC). The corpus has 250,000 running words, comprising research articles, conference proceeding paper, power point lecture presentations and some sections of computer science textbooks.

We set up a four-step criterion for the selection of journal article. Firstly, the article must be a research article focused on empirical study and having identifiable written structures of *Introduction, Method, Result, Discussion,* and *Conclusion* sections. We included conclusion section because in our opinion the section is very important that many researchers highlight their findings and contributions of the studies. Secondly, the article had to be published between 2017 and 2020. The rationale for this is to capture recent development of new vocabulary in the discipline. Thirdly, the chosen research article had to be relevant to each such sub-discipline. Fourthly, the length of the chosen article had to be between 4,000 and 13,000 words long. The rationale was to enable us to access a large number of texts to be compiled. One important point is that we did not consider only articles authored by native speakers of English as a part of our criterion because we believe all the articles selected were from the peer-reviewed journals and thus spelling, as well as grammar had been checked. Having set up the four criteria, we selected 30 research articles from each ten sub-disciplines of Computer Science totaling 300 research articles.

Table 1. Computer Science Sub-disciplines defined by the ACM

	Computer Science Sub-discipline
1	Hardware
2	Information Systems
3	Networks
4	Mathematics of Computing
5	Computing Methodologies
6	Computer Systems Organisation
7	Human-centred Computing
8	Security and Privacy
9	Software and Its Engineering
10	Theory of Computation

3.2 Processing the Data

As noted by Bi (2020) and indeed other scholars extraction of core vocabulary from Computer science literature poses a great difficulty because the literature involves many numerical, programming data, mathematical symbols, figures, tables, and images. We therefore followed three steps to prepare the data. In the first step, we converted all the downloaded research articles from pdf to word document files. We used iLovePDF via this link www.ilovepdf.com to convert all the pdf to word document files. The second step was removing all images, tables, figures, abstracts, author's details, acknowledgements, references, copyright information, funding information, footnotes, and appendices from the chosen research articles. Our third step, was comparing our data against the BNC/COCA 25,000 words family developed by Nation 2017 and Davies 2008. Following Bi's (2020) procedures any items that are not found in the BNC/COCA were thoroughly studied as explained below. Following Nation's (2016) argument that numbers, formulae, non-words, and other forms which contain both mixture of numbers and letters are usually not counted as words, we enclosed such items in triangle brackets (< >) (Bi, 2020). In addition, we used PowerGREP (Goyvaerts, 2016) in searching and processing regular expressions such as [^ a- zA – Z] + [0-9a-zA-Z]*. Hence all these features which did not count as words were removed from the texts. Konstantakis (2010) argues that proper nouns do not contribute any difficulty or burden in learning which can be removed or edited. Following Minshall (2013) procedures, we used this expression \ [. *? \ ] in removing all in-text citations. However, this expression did not remove all the proper names of the in-text citations. We had to delete the remaining proper names manually. We then converted all the texts into TXT files.

Having completed the above steps, and following Sinclair's (2005) argument on corpus development in relation to covering representativeness, we developed a corpus of Computer Science (hereafter CSRAC) of 2,500,990 running words from ten sub-disciplines of Computer Science and also established a sub-corpus for each sub-discipline as shown in table two below.

Table 2. Corpus length of each sub-discipline in tokens

	Sub-discipline	**Corpus length**
1	Hardware	249,989
2	Information Systems	250,120
3	Networks	250,403
4	Mathematics of Computing	250,145
5	Computing Methodologies	250,124
6	Computer Systems Organisation	249,825
7	Human-centred Computing	250,118
8	Security and Privacy	250,174
9	Software and Its Engineering	249,990
10	Theory of Computation	250,102
	Total length	**2,500,990**

3.3 Word Selection Criteria

Having cleared all unwanted data from the CSRAC, we then focused our attention on word selection criteria. In selecting the target words from the CSRAC, we first used Lancsbox (Brezina, et al., 2020). This software is used for natural language processing, such as part-of-speech (POS) tag and to lemmatize words in the raw texts. It is also used for calculation of relative frequency, as well as dispersion. We used the software to tag POS in the CSRAC and enabled us to select lemmatized words. Unlike many previous studies which preferred headword/word family form to lemma form (Coxhead, 2000; Minshall, 2013; Yang, 2015; & Bi, 2020) in this study, we used lemma form for our results. The rationale for choosing a lemma form rather than word family form is a three-fold as explained by (Lei & Liu, 2016). Firstly, a lemma form shows part of speech and word family form does not show part of speech. Secondly, since a lemma form shows part of speech learners could pay more attention to that particular word class which is having a higher frequency and ignore those which are less frequent. Thirdly, a word family form is focused on the dictionary form and learners might be forced to concentrate on a word family even if it is less frequent and ignore the main lemmatized words to be learnt.

Having done the POS-tag, we then used Python software to extract target lemmas from our corpus data (Lei & Liu, 2016). It can be used for many linguistic analyses. As mentioned earlier, we compared our raw data with the BNC/COCA word family lists. The software we used for the comparison was AntWordProfiler (Anthony, 2014). The tool is used for a number of natural language processes, such as analysis of word range, vocabulary frequency, as well as comparing the target data with any other corpora. We then set up a five-criterion for the selection of target words. These criteria were drawn from previous studies (Coxhead, 2000; Wang, et al, 2008; Gardner & Davies, 2014; Lei & Liu, 2016; & Roesler, 2020).

The first criterion for the selection of NCSAWL was minimum frequency. This is defined by Coxhead (2000) as number of individual appearances of word in the corpus. Coxhead (2000) had a corpus of 3.5 million words and used the threshold of 100 times occurrences, which translates to 28.57 times per million words. Lei and Liu (2016) and Wang et al. (2008) used Coxhead's (2008) threshold of 28 times occurrences per one million tokens in their corpus. Roesler (2020) also used the threshold of 100 times in a corpus of 3.5 million tokens. Chen and Lei (2019) used the threshold of 100 times in a corpus of 10 million tokens. Yang (2015) considered threshold of 33 times occurrences which was the one third of Coxhead's (2000) threshold because the corpus had a one million tokens. Bi (2020) used a threshold of 13.31 times per one million words after experimenting different cutoff frequencies. It appears there is a lack of explicit criteria for setting up a threshold frequency. In our study, we did experiment with a number of different threshold frequency points, such as 35 and 40 times per million words. We discovered only a few lexical words can be considered and a lot of words which are having special meanings relevant to computer science were not included. We finally considered occurrences of 70 times in our corpus of 2.5 million tokens similar to Coxhead's (2000) threshold of 28 times per one million running words.

Our second criterion was range, this means that a chosen lemma must appear in a wide range of sub-corpora. Coxhead (2000) and indeed many other studies of academic word list decided that a chosen lemma must occur in at least half of the sub-discipline (50%). However, Gardner and Davies (2014); Lei and Liu (2016) and Roesler (2020) decided to require a more rigorous ratio. For example, Gardner and Davies (2014) required that for a lemma to be selected must have 20% of expected appearance in 7 of their 9 disciplines (ratio of 78%). In this study, however, we decided to require that a lemma to be selected must occur in 5 of the 10 sub-disciplines (50%) similar to Coxhead's (2000) requirement. Our rationale for using Coxhead's (2000) threshold is two-fold. Firstly, many specialized computer words might have not been included in our word list and have many frequencies. Secondly, all the lemmas considered were selected after we checked their meanings and uses in the Computer Science dictionaries and found that they have meanings and uses in that discipline. We felt that since the lemmas have a computer science meaning and use and appeared in 5 out of 10 sub-disciplines, such lemmas should be considered. Here we compromised on a more rigorous ratio similar to Gardner and Davies (2014) and others because of these reasons.

The third criterion was dispersion. Scholars define dispersion as a statistical measure which indicates how frequent a lemma is evenly distributed or spread in the corpus (Gardner & Davies, 2014; Lei & Liu, 2016; Bi, 2020; & Roesler, 2020). Following Gardner and Davies, (2014); Lei and Liu, (2016); and Roesler, (2020), we selected Juilland's D dispersion measure (Juilland, et al., 1970). As reported by many scholars there is a lack of agreement on a specific threshold for the ideal cutoff point because different scholars used different threshold for the cutoff point (Gries, 2019; & Lei & Liu, 2016). For example, Bi (2020) decided to require that for a word to be considered on the list it must have a Juilland's D value of 0.4 and Paquot (2005) considered a cutoff threshold point of 0.5. Oakes and Farrow (2007) and Roesler (2020), used a threshold of Juilland' D value of 0.3; while Gardner and Davies (2014) decided to use a threshold of 0.8. Following this, we first tested whether for a lemma to be considered on our list it should have a Juilland D value of 0.6 and 0.7 and this revealed that many high frequent lemmas which are useful for the discipline of Computer Science were not included. Because we checked computer science dictionary and they have meanings and usages in the computer science. We finally decided to require that for a lemma to be included on our list must have a Juilland's D value of 0.5 as a cutoff point which is similar to Paquot's (2001) threshold.

Discipline measure is our fourth criterion, this measure prevents a lemma to be clustering in a few sub-disciplines. Gardner and Davies (2014) decided to use discipline measure to exclude lemmas which are discipline specifics to only a few disciplines. They stated that the frequency of a lemma must not appear more than 3 times the expected frequency in the 9 sub-sub-corpora. Lei and Liu (2016) used a different threshold that a lemma must not have more than 3 times the expected frequency in more than any three of the twenty-one sub-corpora. Roesler (2020) required that a lemma must not occur more than three times the expected frequency in any 3 of the ten sub-corpora. Following Roesler (2020), in this study, we required that a lemma to be considered on our list must not occur more than three times the expected frequency in any three of the ten sub-corpora.

Our last criterion was special meaning. Unlike previous studies (Lei, & Liu, 2016; & Roesler, 2020), where they considered only general high-frequency words which met the above criteria by checking their special meanings in special dictionaries relevant to the target discipline, we decided to look up the meanings of all the words on our word list of this study. Each lemma included on our final list of computer science academic word list was checked in computer science dictionaries and it was found to have special meaning and use in the discipline. We used three computer science dictionaries to check their meanings, the dictionaries were: Oxford Dictionary of Computer Science (2016), Computer Dictionary (2016), and Computer Science Dictionary (2017). It is important to note that we decided to use these three computer science dictionaries because of the number of entries of each dictionary. For example, a lemma, *baseline* was not found in the Oxford Dictionary of Computer Science (2016), but it appeared in two dictionaries mentioned above, referring to any set of software documents and components which have been reviewed and accepted formally for current production. Another example, *beacon* was only found in Computer Dictionary (2016), having meaning of a device which transmits signal via Bluetooth.

4. Results and Discussion

For the results of step one, we extracted a total of 1,394 lemmas from our corpus as potential words for the NCSAWL. Regarding step two of checking each word whether it appeared in a wide range of sub-corpora, our potential list was reduced to 997 lemmas. This indicates that 397 words were removed, which represents 28.6% from our generated list of 1,394 lemmas. Dispersion was our next step, which again assisted us to ensure that a lemma appeared evenly across the corpora. Our list was further reduced from 997 to 605 lemmas. On our list of 605 lemmas, we identified a total of 366 lemmas that were also on the new-GSL; unlike Lei and Liu (2016) where they randomly checked whether items have special meanings or uses in Medicine, here we checked the meanings and uses of all 366 lemmas in computer science dictionaries. We found that 205 out of 366 lemmas have special meanings and uses in the discipline of Computer Science. For example, *cloud* has different meaning and use in Computer Science from its general meaning and use. When *cloud* is used in Computer Science it means a data center which has a lot of servers to the internet and perform services. However, its general meaning is ‘a visible mass of water which suspended in the air’. In addition, *client* when used in Computer Science means ‘any laptop, desktop or smartphone which sends and receives information from a server’; unlike its general meaning of ‘person who pays for goods and services’; or ‘someone who seeks for the services of a lawyer’. Furthermore, *bucket* in Computer Science means ‘a reserved amount of memory which usually holds a single item or multiple items of data’, but its general meaning is ‘any cylindrical vessel which usually open at the top’. Another example is *cell* when used in Computer Science means ‘the intersection of a raw and column’ or it could mean ‘the storage for one unit of information, generally one character, one byte or one word etc.’ However, its general meaning could mean ‘a room where a prisoner is kept’ or ‘a basic and usually small unit of an organization or movement etc.’

It is evident that these words have special meanings and uses in the discipline of Computer Science, which could not be excluded on the NCSAWL. One important point to make here is that in spite of their usefulness in Computer Science, these words sometimes are also used in their generic meanings as mentioned above. The removal of 161 lemmas on our list of 605 lemmas brings to a total of 444 lemmas as our final list of NCSAWL (see appendix 1).

4.1 Comparing NAWL and new-GSL on NCSAWL

Following practices of Gardner and Davies (2014), Lei and Liu (2016) and Yang (2015), we calculated and compared the coverage of our lemma-based NCSAWL with Gardner and Davies (2014) NAWL and Brezina and Gablasova’s (2015) new-GSL. Since both the NAWL and new-GSL are lemma-based, we did not follow Brezina and Gablasova’s (2015) three steps of calculating and comparing our word list with lemma-based and word family lists. We first compared and calculated our lemma-based of 444 words with Gardner and Davies’ (2014) NAWL. The result shows that a total of 116 words overlapped between our list and the NAWL, which represents 26.1% of our list, unlike Roesler (2020) who reported an overlap of 37% between CSAVL and NAWL.

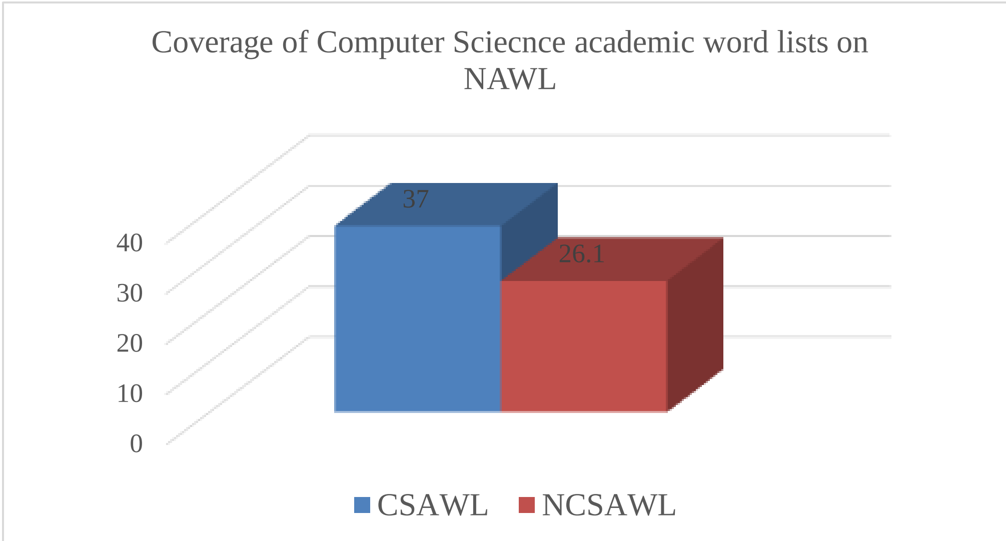

Figure 1. Coverage of Computer Science Word lists on NAWL

As can be seen in figure 1 above, the coverage of our NCSAWL and CSAWL on NAWL has some kind of variance as mentioned above, in that our list has 26.1% overlap with the NAWL; whereas the CSAWL has an overlap of 37% with the NAWL. This difference between our list and the CSAWL might be associated with the number of lemmas on each list, because our list is almost fifty percent of the total number of lemmas on CSAWL. Another possible cause of the difference is the size between our corpus and that of CSAWL corpus. The former corpus has 2.5 million words and the latter corpus has 3.5 million running words. This might be one of the possible reasons for such difference.

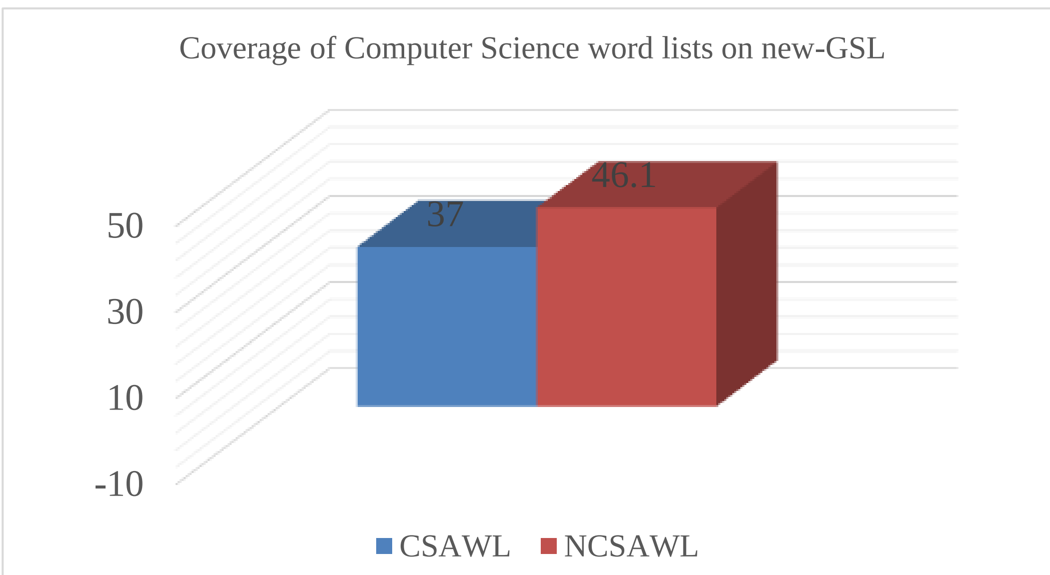

Figure 2. Coverage of Computer Science word lists on new-GSL

As can be seen in figure 2 above, we also compared our list with the new-GSL as mentioned above, 205 (46.1%) out of 444 lemmas overlapped. Again, the CSAWL has 37% overlap with the new-GSL. One striking finding is the variance of percentage between our list and that of CSAWL, even though our list has 444 lemmas and the CSAWL has 904 lemmas. However, our result is almost consistent with the finding of Lei and Liu (2016) where 45.7% overlap was reported between MAVL and new-GSL.

4.2 Comparing NCSAWL with CSAWL

As mentioned above, following practices of Gardner and Davies (2014) and Lei and Liu (2016), we checked and established representativeness and viability of our NCSAWL by calculating and comparing the coverage of NCSAWL across general, academic and computer science corpora. We used the British National Corpus (the BNC) as general corpus and we also used a sub-corpus of academic writing of the BNC as our academic corpus, CSRAC and CSC were used as computer science corpora. As can be seen in table 3 below, our NCSAWL has a coverage of 3.1% in the general corpus of the BNC and it has a coverage of 5.80% in the BNC academic corpus. Our

NCSAWL has also covered CSRAC by 20.33% and CSC 19.08% respectively. This result is almost consistent with the findings of Lei and Liu (2016) where 3.69% was reported for comparing and coverage of MAVL in the BNC and 6.65%, 19.44% and 20.18% were also reported for BNC academic, MAEC and MTEC corpora respectively. However, our results have higher coverage compared to Roesler's (2020) finding, we will discuss further in the next section. Indeed, our results show that the NCSAWL is a specialized academic word list of computer science as argued by Lei and Liu (2016) that if the coverage of the NCSAWL is higher in computer science corpora than general and academic corpora, then the list contains words which are more frequently used in computer science rather than general academic English.

Table 3. Coverage of NCSAWL across general, academic, and computer science corpora

	BNC	**BNC Academic**	**CSRAC**	**CSC**
NCSAWL	3.1%	5.80%	20.33%	19.08%

Regarding the comparison of NCSAWL with CSAWL, unlike previous studies of comparing and calculating coverage of different word lists in the same specialized corpora, we did follow a different approach. Since both the NCSAWL and CSAWL developed two different corpora (primary and supplementary) and both word lists were lemma-based, we are of the opinion that it is necessary to report and compare the results of each word list, because by using one of the corpora developed from either one of the studies might have favoured one of the studies. As can be seen in table 4 below, the NCSAWL has more coverage in all the corpora compared to the CSAWL. It is interesting to note that in both the primary and supplementary corpora the NCSWAL has more coverage than the CSAWL. For example, in the BNC, 3.1% and 2.96% were reported for NCSAWL and CSAWL; for the BNC academic 5.80% and 4.93% both word lists were recorded; and 20.33%, 19.08%; and 16.87% and 16.06% of primary and supplementary corpora were also reported for both NCSAWL and CSAWL as shown in table 4 below. It is evident that the NCSAWL is robust and all the words occurred in the supplementary corpus (CSC). Unlike the CSAWL, the NCSAWL is specific not general and the list contains fewer items. It could be possible that students might learn better from the NCSAWL than CSAWL.

Table 4. Comparing NCSAWL with CSAWL

	BNC	**BNC Academic**	**CSRAC/CSAC1= Primary corpora**	**CSC/CSAC2= Supplementary corpora**
NCSAWL	3.1%	5.80%	20.33%	19.08%
CSAWL	2.96%	4.93%	16.87%	16.06%

5. Conclusion

In this study, we reported the development of a new computer science academic word list (NCSAWL) by drawing on and combining previous procedures and contemporary studies on establishment of academic word lists. On the basis of different comparative analyses, the NCSAWL has a much better coverage of computer science English and it would probably better serve the computer science students. As shown above, our list has 46.1% coverage of the new-GSL and 26.1% of the NAVL respectively. The coverage of our list on computer science research corpus is 20.33% and the supplementary computer science for validity test is 19.08%. This clearly shows a wider coverage of our list based on computer science research article corpus when compared to the previous computer science word list.

6. Pedagogical Implications

The NCSAWL is relatively short and has a high coverage of computer science texts. It has numerous implications for computer science learners, English teachers, researchers, as well as material writers and course syllabus designers who are working in the discipline of computer science. For example, English computer teachers should focus on teaching learners the most high-frequent words which have a dispersed coverage and have special meaning and use in the discipline of computer science. Teachers could also raise the awareness of learners that some words have different meanings and uses in general English. The material designers for English for academic and special purposes could incorporate the NCSAWL vocabulary into their academic reading and writing materials for computer science students. Researchers and English language teachers who are interested in expanding their computer science academic vocabulary could also use this NCSAWL.

7. Limitations and Future Research

This study has many limitations. One of its limitations is that the word list is produced on only single word unit, while many scholars argue that multi-word unit is very important for second language learners (Bi, 2020; & Martinez & Schmitt, 2012). The second limitation is that the corpus was developed from only research articles. Future research should incorporate other genres of computer science to have more representativeness and perhaps also include multi-word units.

Declaration of conflicting interests

The authors declared no potential conflicts of interest in relation to the research, authorship and publication of this article

References

Anthony, L. (2014). *AntWordProfiler (version 1.4.1) [computer software]*. Retrieved from http://www.laurenceanthony.net/software/antwordprofiler/

Bi, J. (2020). How large a vocabulary do Chinese computer science undergraduates need to read English-medium specialist textbooks? *English for Specific Purposes*, *58*, 77-89. https://doi.org/10.1016/j.esp.2020.01.001

Brezina, V., & Gablasova, D. (2015). Is there a core general vocabulary? Introducing the new general service list. *Applied Linguistics*, *36*(1), 1-22. http:/doi:10.1093/applin/amt018

Browne, C. (2014). A new general service list: The better mousetrap we've been looking for. *Vocabulary Learning and Instruction*, *3*(2), 1-10. https://doi.org/10.7820/vli.v03.2.browne

Butterfield, A., Ngondi, G. E., & Kerr, A. (Eds.). (2016). *A dictionary of computer science*. Oxford University Press. https://doi.org/10.1093/acref/9780199688975.001.0001

Chen, H., & Lei, G. (2019). Developing a Technical Words List for Research Articles in Computer Science Discipline. *English Language Teaching*, *12*(10), 131-141. https://doi.org/10.5539/elt.v12n10p131

Chen, Q., & Ge, G. C. (2007). A corpus-based lexical study on frequency and distribution of Coxhead's AWL word families in medical research articles (RAs). *English for Specific Purposes*, *26*(4), 502-514. https://doi.org/10.1016/j.esp.2007.04.003

Chung, T. M., & Nation, P. (2003). Technical vocabulary in specialised texts. *Reading in a foreign language*, *15*(2), 103.

Cobb, T., & Horst, M. (2004). Is there room for an academic word list in French. *Vocabulary in a second language*, *23*, 15-38. https://doi.org/10.1075/lllt.10.04cob

Corson, D. (1997). The learning and use of academic English words. *Language learning*, *47*(4), 671-718. https://doi.org/10.1111/0023-8333.00025

Coxhead, A. (2000). A New Academic Word List. *TESOL Quarterly, 34*(2), 213-238. https://doi.org/10.2307/3587951

Coxhead, A., & Demecheleer, M. (2018). Investigating the technical vocabulary of plumbing. *English for Specific Purposes*, *51*, 84-97. https://doi.org/10.1016/j.esp.2018.03.006

Coxhead, A., & Hirsch, D. (2007). A pilot science-specific word list. *Revue française de linguistique appliquée*, *12*(2), 65-78. https://doi.org/10.3917/rfla.122.0065

Davies, M. (2008). *The corpus of contemporary American English: 560 million words, 1990-present.* Retrieved from http://corpus.byu.edu/coca/

Farrell, P. (1990). A lexical analysis of the English of electronics and a study of semi-technical vocabulary (CLCS Occasional Paper No. 25). Dublin: Trinity College. (ERIC Document Reproduction Service No. ED332551

Gardner, D., & Davies, M. (2014). A new academic vocabulary list. *Applied linguistics*, *35*(3), 305-327. https://doi.org/10.1093/applin/amt015

Gilner, L. (2011). A primer on the General Service List. *Reading in a foreign language*, *23*(1), 65-83.

Goldenberg, C. (2008). Teaching English Language Learners: What the research does— and does not—say. *American Educator*, *Summer*, 8-44.

Goyvaerts, J. (2016). *PowerGREP (version 4.7.3) [computer software].* Retrieved from http://www.powergrep.com/

Gries, S. Th. (2019). Analyzing dispersion. In S. Th. Gries & M. Paquot (Eds.), *Practical Handbook of Corpus Linguistics*. Retrieved from http://www.stgries.info/research/ToApp_STG_Dispersion_PHCL.pdf

Hirsh, D., & Nation, P. (1992). What vocabulary size is needed to read unsimplified texts for pleasure? *Reading in a foreign language, 8,* 689-696.

Hsu, W. (2013). Bridging the vocabulary gap for EFL medical undergraduates: The establishment of a medical word list. *Language Teaching Research*, *17*(4), 454-484. https://doi.org/10.1177/1362168813494121

Huang, J. Y. (2007). Exploring the use of vocabulary from academic word list in applied linguistic articles. *Unpublished master's thesis. Taiwan: National Tsing Hua University*.

Hyland, K., & Tse, P. (2007). Is there an "academic vocabulary"? *TESOL quarterly*, *41*(2), 235-253. https://doi.org/10.1002/j.1545-7249.2007.tb00058.x

Juilland, A. G., Brodin, D. R., & Davidovitch, C. (1970). *Frequency dictionary of French words* (Vol. 1). Mouton

Khani, R., & Tazik, K. (2013). Towards the development of an academic word list for Applied Linguistics research articles. *RELC Journal*, *44*(2), 209-232. https://doi.org/10.1177/0033688213488432

Konstantakis, N. (2010). *Constructing a word list for the domain of business.* (Unpublished PhD thesis), Swansea University, Swansea.

Laufer, B. (1990). Ease and difficulty in vocabulary learning: Some teaching implications. *Foreign Language Annals*, *23*(2), 147-155. https://doi.org/10.1111/j.1944-9720.1990.tb00355.x

Laufer, B. (1992). Reading in a foreign language: How does L2 lexical knowledge interact with the reader's general academic ability'. *Journal of Research in Reading*, *15*(2), 95-103. https://doi.org/10.1111/j.1467-9817.1992.tb00025.x

Lei, L., & Liu, D. (2016). A new medical academic word list: A corpus-based study with enhanced methodology. *Journal of English for* *academic purposes*, *22*, 42-53. https://doi.org/10.1016/j.jeap.2016.01.008

Mart ńez, I. A., Beck, S. C., & Panza, C. B. (2009). Academic vocabulary in agriculture research articles: A corpus-based study. *English for specific purposes*, *28*(3), 183-198. https://doi.org/10.1016/j.esp.2009.04.003

Martinez, R., & Schmitt, N. (2012). A phrasal expressions list. *Applied linguistics*, *33*(3), 299-320. https://doi.org/10.1093/applin/ams010

McEnery, T., & Hardie, A. (2011). *Corpus linguistics: Method, theory and practice*. Cambridge University Press. https://doi.org/10.1017/CBO9780511981395

Minshall, D. E. (2013). *A computer science word list* (Master's thesis). Retrieved from https://www.baleap.org/wp-content/uploads/2016/03/Daniel-Minshall.pdf

Nagy, W., & Townsend, D. (2012). Words as tools: Learning academic vocabulary as language acquisition. *Reading Research Quarterly*, *47*(1), 91-108. https://doi.org/10.1002/RRQ.011

Nation, I. (2006). How large a vocabulary is needed for reading and listening? *Canadian modern language review*, *63*(1), 59-82. https://doi.org/10.3138/cmlr.63.1.59

Nation, I. S. (2001). *Learning vocabulary in another language*. Ernst Klett Sprachen. https://doi.org/10.1017/CBO9781139524759

Nation, I. S. (2013). *Teaching & learning vocabulary*. Boston: Heinle Cengage Learning.

Nation, I. S. P. (2004). A study of the most frequent word families in the British National Corpus. *Vocabulary in a second language: Selection, acquisition, and testing*, *10*, 3-13. https://doi.org/10.1075/lllt.10.03nat

Nation, I. S. P. (2017). The BNC/COCA Level 6 word family lists (Version 1.0.0). Retrieved from http://www.victoria.ac.nz/lals/staff/paul-nation

Nation, P., & Kyongho, H. (1995). Where would general service vocabulary stop and special purposes vocabulary begin? *System*, *23*(1), 35-41. https://doi.org/10.1016/0346-251X(94)00050-G

Naton, I. S. P. (2016). *Making and using word lists for language learning and testing.* John Benjamins. https://doi.org/10.1075/z.208

Oakes, M. P., & Farrow, M. (2007). Use of the chi-squared test to examine vocabulary differences in English language corpora representing seven different countries. *Literary and Linguistic Computing*, *22*(1), 85-99. https://doi.org/10.1093/llc/fql044

Paquot, M. (2001). Towards a productively-oriented academic word list. *higher education*, *187*, 216. http://hdl.handle.net/2078.1/76045

Quero, B. (2015). Estimating the vocabulary size of L1 Spanish ESP learners and the vocabulary load of medical textbooks. Unpublished PhD Thesis, Victoria University of Wellington, New Zealand.

Richards, J. C. (1974). Word lists: problems and prospects. *RELC journal*, *5*(2), 69-84. https://doi.org/10.1177/003368827400500207

Roesler, D. (2020). *A Computer Science Academic Vocabulary List*. https://doi.org/10.1016/j.jeap.2021.101044

Schmitt, N., Jiang, X., & Grabe, W. (2011). The percentage of words known in a text and reading comprehension. *The Modern Language Journal*, *95*(1), 26-43. https://doi.org/10.1111/j.1540-4781.2011.01146.x

Shaw, P. (1991). Science research students' composing processes. *English for specific purposes*, *10*(3), 189-206. https://doi.org/10.1016/0889-4906(91)90024-Q

Thurstun, J., & Candlin, C. (1998). Exploring academic English. *Macquarie University: NCELTR*.

Todd, R. W. (2017). An opaque engineering word list: Which words should a teacher focus on? *English for Specific Purposes, 45*, 31-39. https://doi.org/10.1016/j.esp.2016.08.003

Tongpoon-Patanasorn, A. (2018). Developing a frequent technical words list for finance: A hybrid approach. *English for Specific Purposes*, *51*, 45-54. https://doi.org/10.1016/j.esp.2018.03.002

Valipouri, L., & Nassaji, H. (2013). A corpus-based study of academic vocabulary in chemistry research articles. *Journal of English for Academic Purposes*, *12*(4), 248-263. https://doi.org/10.1016/j.jeap.2013.07.001

Wang, J., Liang, S. L., & Ge, G. C. (2008). Establishment of a medical academic word list. *English for Specific Purposes*, *27*(4), 442-458. https://doi.org/10.1016/j.esp.2008.05.003

Ward, J. (2009). A basic engineering English word list for less proficient foundation engineering undergraduates. *English for specific purposes*, *28*(3), 170-182. https://doi.org/10.1016/j.esp.2009.04.001

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

Xue, G., & Nation, I. S. P. (1984). A university word list. *Language learning and communication*, *3*(2), 215-229.

Yang, M. N. (2015). A nursing academic word list. *English for specific purposes*, *37*, 27-38. https://doi.org/10.1016/j.esp.2014.05.003

Appendix: The New Computer Science Academic Word List

Notes: A letter ("adj" for adjective, "adv" for adverb, "n" for noun, and "v" for verb) is given to each word indicating the part of speech being referenced.

1	Access_n	41	Board_n	81	Comment_n
2	Acknowledge_v	42	Bookmark_v	82	Compile_n
3	Actor_n	43	Bottleneck_n	83	Component_n
4	Address_n	44	Bottom_n	84	Compress_v
5	Agent_n	45	Box_n	85	Computer_n
6	Allocate_v	46	Branch_n	86	Concatenation_n
7	Anomaly_n	47	Break_v	87	Configuration_n
8	App_n	48	Brightness_n	88	Connect_n
9	Approach_n	49	Browse_v	89	Contact_n
10	Architecture_n	50	Bucket_n	90	Container_n
11	Argument_n	51	Buffer_n	91	Content_n
12	Array_n	52	Bug_n	92	Contrast_n
13	Assemble_v	53	Build_n	93	Control_n
14	Associate_n	54	Bus_n	94	Cryptography_n
15	Asymmetric_adj	55	Byte_n	95	Customize_v
16	Asynchronous_adj	56	Cable_n	96	Cut_v
17	Atom_n	57	Cache_n	97	Data_n
18	Attach_v	58	Calibrate_v	98	Database_n
19	Attack_n	59	Call_n	99	Datacenter_n
20	Attribute_n	60	Capture_v	100	Debug_v
21	Audio_n	61	Card_n	101	Decode_v
22	Authorization_n	62	Cell_n	102	Decoupling_n
23	Automate_v	63	Chain_n	103	Decrypt_v
24	Background_n	64	Channel_n	104	Default_n
25	Backup_n	65	Character_n	105	Demonstration_n
26	Band_n	66	Chip_n	106	Desktop_n
27	Bandwidth_n	67	Chrome_n	107	Developer_n
28	Bank_n	68	Circuit_n	108	Device_n
29	Barcode_n	69	Class_n	109	Digital_adj
30	Base_n	70	Clear_n	110	Disk_n
31	Baseline_n	71	Click_v	111	Display_n
32	Batch_n	72	Client_n	112	Document_n
33	Beacon_n	73	Clock_n	113	Domain_n
34	Benchmark_n	74	Clone_n	114	Down_adv
35	Bias_n	75	Cloud_n	115	Download_n
36	Bin_n	76	Cluster_n	116	Drag_v
37	Binary_n	77	Cold_n	117	Drive_n
38	Bind_v	78	Collector_n	118	Element_n
39	Biometric_n	79	Column_n	119	Encode_v
40	Block_n	80	Command_n	120	Encryption_n

121	Enter_v	161	Instrument_v	201	Manufacture_n
122	Entry_n	162	Interactive_adj	202	Mark_n
123	Erase_v	163	Interface_n	203	Mask_n
124	Event_n	164	Internet_n	204	Master_n
125	Exception_n	165	Interpolation_n	205	Match_n
126	Explorer_n	166	Interpret_n	206	Matrix_n
127	Extraction_n	167	Interrupt_v	207	Maximize_v
128	Feed_n	168	Intersect_v	208	Media_n
129	Fetch_v	169	Intruder_n	209	Memory_n
130	Field_n	170	Isolation_n	210	Merge_v
131	File_n	171	Iteration_n	211	Message_n
132	Filter_v	172	Job_n	212	Metadata_n
133	Firmware_n	173	Join_n	213	Microcontroller_n
134	Flag_n	174	Journal_n	214	Migrate_v
135	Flash_n	175	Jump_n	215	Minimize_v
136	Flush_v	176	Justify_v	216	Mode_n
137	Focus_n	177	Key_n	217	Model_n

138	Fold_n	178	Keyboard_n	218	Modular_n
139	Form_n	179	Label_n	219	Module_n
140	Forth_n	180	Laptop_n	220	Monitor_n
141	Frame_n	181	Latency_n	221	Mouse_n
142	Gate_n	182	Launch_v	222	Move_n
143	Get_n	183	Library_n	223	Multiplex_v
144	Google_v	184	Line_n	224	Navigate_v
145	Granularity_n	185	Link_n	225	Network_n
146	Guard_n	186	List_n	226	Node_n
147	Hardware_n	187	Literal_n	227	Noise_n
148	Hash_n	188	Load_v	228	Normalization_n
149	Head_n	189	Local_n	229	Null_n
150	Header_n	190	Log_n	230	Number_n
151	Hierarchy_n	191	Logic_n	231	Object_n
152	History_n	192	Login_n	232	Offset_n
153	Hit_n	193	Long_n	233	Online_adj
154	Host_v	194	Look_n	234	Open_n
155	Idle_adj	195	Loop_n	235	Optimal_adj
156	Implement_v	196	Machine_n	236	Orthogonal_adj
157	Index_n	197	Make_n	237	Output_n
158	Inheritance_n	198	Maintainer_n	238	Overhead_n
159	Input_n	199	Malicious_adj	239	Overlap_v
160	Install_v	200	Manual_adj	240	Pack_v

241	Page_n	281	Put_v	321	Scan_n
242	Pair_v	282	Quantify_v	322	Schema_n
243	Paradigm_n	283	Query_v	323	Scope_n
244	Parameter_n	284	Queue_n	324	Screen_n
245	Parent_n	285	Radio_n	325	Script_n
246	Park_v	286	Rank_n	326	Search_v
247	Partition_n	287	Read_v	327	Search_n
248	Pass_n	288	Real_adj	328	Sector_n
249	Password_n	289	Reconfiguration_n	329	Seek_v
250	Patch_n	290	Record_n	330	Segment_n
251	Path_n	291	Recover_v	331	Select_v
252	Payload_n	292	Recursion_n	332	Self_n
253	Perform_n	293	Reflection_n	333	Sensor_n
254	Persistence_n	294	Register_v	334	Server_n
255	Perspective_n	295	Release_n	335	Service_n
256	Pilot_n	296	Remote_n	336	Session_n
257	Pixel_n	297	Rename_n	337	Set_n
258	Platform_n	298	Rep_v	338	Shift_n
259	Plot_n	299	Replace_n	339	Signal_n
260	Pop_v	300	Report_n	340	Simulation_n
261	Port_v	301	Repository_n	341	Sleep_n
262	Pose_n	302	Reset_v	342	Slice_n
263	Post_v	303	Resiliency_n	343	Smart_adj
264	Power_n	304	Resolution_n	344	Software_n
265	Practise_v	305	Resolve_v	345	Solutions_n
266	Predicate_n	306	Resource_n	346	Sort_v
267	Preemption_n	307	Response_n	347	Sound_n
268	Print_v	308	Retrieval_n	348	Source_n
269	Probe_n	309	Return_n	349	Space_n
270	Procedure_n	310	Reuse_v	350	Speed_v
271	Processor_n	311	Robot_n	351	Spreadsheet_n
272	Profile_n	312	Robust_adj	352	Stack_n
273	Programme_n	313	Root_n	353	State_n
274	Programming_n	314	Route_v	354	Static_adj
275	Propagation_n	315	Router_n	355	Station_n
276	Protocol_n	316	Run_v	356	Store_v
277	Prototype_n	317	Runtime_n	357	Stream_v
278	Publish_v	318	Sample_v	358	String_n

279	Pulse_n	319	Save_v	359	Structure_n
280	Push_v	320	Scan_v	360	Subject_n

361	Subset_n	403	Trigger_v
362	Subsystem_n	404	Trust_n
363	Supervisor_n	405	Tuple_n
364	Support_v	406	Turn_n
365	Surface_n	407	Type_v
366	Swap_n	408	Type_n
367	Switch_v	409	Union_n
368	Symbol_n	410	Unique_n
369	Symmetric_adj	411	Unit_n
370	Synchronise_v	412	Up_adj
371	Syntax_n	413	Update_v
372	Synthesization_n	414	Update_n
373	System_n	415	Upload_v
374	Table_n	416	User_n
375	Tag_n	417	Utility_n
376	Tap_v	418	Utilization_n
377	Task_n	419	Valid_adj
378	Technique_n	420	Value_n
379	Test_n	421	Variable_n
380	Theme_n	422	Version_n
381	Threat_n	423	Video_n
382	Threshold_n	424	View_n
383	Throughput_n	425	Virtual_adj
384	Timestamp_n	426	Visit_n
385	Token_n	427	Voice_n
386	Tool_n	428	Voltage_n
387	Topology_n	429	Volume_n
388	Touch_n	430	Walk_v
389	Trace_v	431	Wall_n
390	Trace_n	432	Wave_n
391	Track_v	433	Web_n
392	Traffic_n	434	Website_n
393	Train_n	435	While_n
394	Transaction_n	436	Wifi_n
395	Transcribe_v	437	Wikipedia_n
396	Transfer_v	438	Window_n
397	Transitor_n	439	Windows_n
398	Translate_v	440	Wire_n
399	Transmit_v	441	Wireless_n
400	Transport_v	442	Word_n
401	Tree_n	443	Write_v
402	Triangle_n	444	Yield_v