Frequency Consolidation Among Word N-Grams
A Practical Procedure

Andreas Buerki(✉)

Centre for Language and Communication Research,
Cardiff University, Cardiff, Wales, UK
buerkiA@cardiff.ac.uk

Abstract. This paper considers the issue of frequency consolidation in lists of different length word n-grams (i.e. recurrent word sequences) extracted from the same underlying corpus. A simple algorithm – enhanced by a preparatory stage – is proposed which allows the consolidation of frequencies among lists of different length n-grams, from 2-grams to 6-grams and beyond. The consolidation adjusts the frequency count of each n-gram to the number of its occurrences minus its occurrences as part of longer n-grams. Among other uses, such a procedure aids linguistic analysis and allows the non-inflationary counting of word tokens that are part of frequent n-grams of various lengths, which in turn allows an assessment of the proportion of running text made up of recurring chunks. The proposed procedure delivers frequency consolidation and substring reduction among word n-grams and is independent of any particular method of n-gram extraction and filtering, making it applicable also in situations where full access to underlying corpora is unavailable.

Keywords: Multiword expressions · Word n-grams · Corpus linguistics

1 Introduction

The present paper presents a procedure for frequency consolidation among lists of recurrent word sequences (i.e. word n-grams) of various lengths, as implemented in the software programme SubString [1]. Word sequences that occur again and again in largely the same form are often referred to as multi-word expressions (MWEs) and have been of considerable interest to corpus linguistics as well as natural language processing (NLP). In corpus linguistics, the availability of large language corpora and their machine-assisted processing led to the realisation that recurrent sequences are far more widespread than would be predicted on the basis of a model of linguistic knowledge that consists of a store of atomic items (words in the lexicon) and combinatory rules (the grammar). This in turn led to alternative models of linguistic knowledge and processing being proposed such as Sinclair's Idiom Principle [2], Pattern Grammar [3] or Lexical Priming [4]. The existence of large quantities of MWEs in language also influences and supports

R. Mitkov (Ed.): Europhras 2017, LNAI 10596, pp. 432–446, 2017.
https://doi.org/10.1007/978-3-319-69805-2_30

constructionist approaches to grammar (e.g. [5–7], and research in the field of formulaic language, which has attracted a great deal of recent research effort (for a recent overview see [8]). In NLP, MWEs are key to the success of many tasks including machine translation, speech synthesis or term extraction (e.g. [9–11]).

Methods of MWE extraction out of corpus material have therefore been an area of very significant research activity (cf. overviews in [12–14]). The present paper concerns a different but related task that has, by comparison, received scant attention to date. It is a specialized task but one that is key to challenges encountered in a number of situations as discussed below.

Frequency consolidation is applied where there are n-grams that are substrings of other n-grams: the frequency of occurrence of a shorter string is then reduced by the frequency of its longer superstring(s). Frequency consolidation thus works on the basis of extracted and filtered lists of MWEs and provides insight into the frequency structure of those MWEs. A simple case is illustrated in Fig. 1, where the two 3-grams *a nice day* and *have a nice* have their frequencies (in square brackets) consolidated with the 4-gram *have a nice day*. This results in the substrings *have a nice* and *a nice day* appearing with consolidated frequencies of 4 and 1 respectively since the six occurrences as part of the superstring *have a nice day* are removed. Substring reduction additionally occurs if a string receives a consolidated frequency count of zero (or below a certain cut-off frequency), resulting in its deletion and consequently a reduction in the number of substring types occurs.

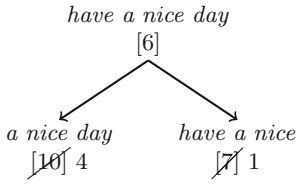

Fig. 1. A simple case of a frequency consolidation between word n-grams

An accurate frequency consolidation and substring reduction procedure is a prerequisite for an empirical assessment of the degree to which running text consists of MWEs; unless frequencies of different length MWEs are consolidated, no accurate figures of the proportion of running words that form part of MWEs can be derived. Although, as indicated above, this is a point of significant theoretical weight and is furthermore thought to vary notably between text types [2, p. 114], [15, p. 29-17], assessments of MWE-density (save for [18], see discussion below) have either relied on estimates based on a single length of n-grams (e.g. [16, pp. 993–997], [19, p. 67ff]), or have had to limit the amount of source material considered to manually countable amounts of text [17,20,21], thus severely limiting the empirical base on which conclusions are drawn. Clearly, neither of these options is satisfactory. An automatic frequency consolidation addresses this problem by

making it easy to determine (and contrast) MWE-density across large amounts of language data. Further, in linguistic analyses of MWEs of various lengths it is often useful to have access to consolidated frequencies of a cluster of MWEs under investigation as this facilitates meaningful comparisons between MWEs of different lengths – an automatic consolidation procedure delivers such results quickly and accurately and in so doing simplifies and speeds up the task of linguistic analysis. Beside these two principal uses, frequency consolidation also reduces the number of n-grams that need to be managed or investigated in a study by eliminating redundant substrings. A word n-gram frequency consolidation procedure is therefore a useful tool when dealing with MWEs and their analysis for linguistic purposes.

The remainder of this paper is structured as follows: in Sect. 2, we take a look at earlier approaches to similar tasks. Then, in Sect. 3, a procedure to deal with frequency consolidation is proposed and documented in detail. In Sect. 4, the proposed procedure is evaluated using corpus data. Conclusions are presented in a final section. Since the procedure is in principle applicable to any type of word sequence (or indeed sequences of other elements) rather than specifically to MWEs, which are more narrowly defined, the general term word n-gram will be used to refer to sequences of words that make up the input to the procedure.

2 Related Work

In a pioneering early study, Altenberg and Eeg-Olofsson [22] used a frequency consolidation procedure involving token-indexation as part of their MWE extraction procedure to ascertain the proportion of recurrent word n-grams in the 500,000 word London-Lund Corpus (results presented in [18]). Procedures using indexation have been discussed since (e.g. [23,24, pp. 147–149]) and remain an important approach to the task of frequency consolidation. A special case of indexation are approaches based on suffix arrays [25] where n-gram extraction and the picking out of interesting n-grams (and possibly the discarding of others) are in effect queries to a corpus, converted to an indexed data structure. As far as could be ascertained, no frequency consolidation procedure of the sort outlined has been suggested for data stored in suffix arrays, although this would certainly be a possibility. Further, a Serial Cascading Algorithm was proposed by Smith (reported in [24, pp. 149–153]). It takes two passes over a corpus to extract, filter and then consolidate n-grams of various lengths. This approach, and indexation-based ones, are integral parts of MWE extraction procedures, that is, they necessarily take as input a corpus of texts, rather than a list of extracted n-grams and their frequencies. This has the advantage of producing results that are maximally faithful to the original context in which n-grams occur, resulting in high accuracy. The disadvantage is a loss of flexibility as these procedures cannot be applied in situations where full access to the underlying corpus material is unavailable, as is frequently the case, for example with the vast Google n-grams corpora [26], n-gram lists made available at the COCA website [27], or with corpora accessible only through corpus portals like the

Sketch Engine [28] or Wmatrix [29] which allow the creation of n-gram lists but not a full frequency consolidation (cf. discussion below).

Among approaches that are independent of a particular extraction procedure and can work with n-gram lists as input, Lü et al. [30] proposed a statistical substring reduction algorithm for fast and accurate 'removal of equal frequency n-gram substrings from an n-gram set' [30, p. 1]. Wible and Tsao [31, p. 29] proposed a similar procedure (referred to as horizontal pruning), where, in addition to the deletion of substrings that match the frequencies of their superstrings, substrings are also deleted if their frequencies are higher than those of the superstring(s), up to a certain maximum 'threshold proportion' [31, p. 29]; this results in a higher number of eliminated substrings. However, both approaches leave unconsolidated the frequencies of substrings that are not eliminated. Consequently, these procedures are suitable only for reducing the number of redundant material in data.

Example 1. (rendering of substrings in Sketch Engine)

settlements in the West Bank 5
... settlements in the West 5
... ... in the West 21
... ... in the 6,348
... ... the West 70
... ... settlements in the 7
... ... in the 6,348
... ... settlements in 9

The n-gram extraction function of the Sketch Engine [28] offers the option to 'hide/nest sub n-grams' [32]. As illustrated using the output shown in Example 1 below, this option groups any sub- (and sub-sub, etc.) strings under the longest extracted n-grams. Crucially, however, no adjustments to frequencies of substrings are made, so even though in Example 1, the substring *settlements in the West* does not occur outside of *settlements in the West Bank*, the substring is still listed with a frequency of 5. When hidden, all substrings (preceded by dots) are removed from lists, even when substrings are much more frequent than the top-level superstring, as illustrated by the string *the West*. Although this aids comparisons of MWEs of different lengths to some degree, the hiding of all substrings, even if more frequent than longer superstrings, and the lack of a consolidation of frequencies limits the usefulness of this approach – its strength lies mainly in providing analysts with a usefully re-arranged view of n-gram lists. The same appears to be case for the option to produce 'collapsed grams' within Wmatrix [29]: although currently this functionality is switched off, it is described as producing 'a tree structure with the longest n-grams on the left and shortest n-grams on the right'.

Other procedures approach the task of handling substrings from the point of view of finding the ideal length of an MWE in a cluster of word sequences that share a common core. Kita et al. [33] documented a procedure which assigns a cost measure to different length word n-grams (cf. also [34,35]). This results in the ideal (according to the measure) extent of an MWE receiving the highest

score while leaving shorter or longer forms with lower scores. Sequences with lower scores are potentially eliminated depending on the threshold value set. A more recent proposal in this paradigm is put forward by Gries [36] and Gries and Mukherjee [37]: first, collocation strength for n-grams of different lengths is measured using Daudaravicius and Marcinkeviciene's gravity measure G [38]. Subsequently, the G-value of each extracted n-gram, starting with 2-grams, is compared to that of its immediate superstring(s) (i.e. n-grams that are one word longer). If the G-values of the superstrings are higher than that of the substring, the substring is removed, otherwise it is retained as a legitimate n-gram despite the existence of larger superstrings. Gries and Wahl propose a procedure involving 'the successive merging of bigrams to form word sequences of various lengths' [39]; while results depend on the setting of a sensible threshold number of successive merges (and this is likely difficult to get right), Gries and Wahl demonstrate using human ratings that MWEs resulting from early merges (vs. late merges) are more often rated as good MWEs. Wible and Tsao [31] documented a similar procedure making use of a normalized MI score. Approaches of this type are useful for allowing 'the length of each n-gram to emerge, as it were, from the data' [37, p. 522]. However, the output, while providing a filtered set of n-grams, does not provide consolidated frequencies for remaining n-grams and is therefore a slightly different task to the one discussed in this paper. Naturally, full access to the underlying corpus data are also required.

In summary, previous research has identified ways of dealing with aspects of substring reduction and frequency consolidation among word n-grams that lead to reductions in redundant substrings and the identification of the ideal length of an n-gram in a cluster as well as full frequency consolidations for cases where full access to the input text corpora is available. However, as far as could be ascertained, no procedure has been suggested to date that covers the uses outlined at the beginning and is sufficiently flexible to cope with situations where only n-gram lists are available as input. Such a procedure will be outlined in the next section.

3 The Procedure

To illustrate how the proposed procedure handles frequency consolidation among different length word n-grams, let us assume we have as input the n-grams given in Example 2 below. These will have been extracted from a corpus and their frequencies of occurrence in the corpus are indicated by the number following each n-gram.

Example 2. (example input to a frequency consolidation):

have a lovely time	15
have a lovely	58
a lovely time	44
have a	37,491
a lovely	101
lovely time	44

Example 3. (consolidated output):

have a lovely time	15
have a lovely	43
a lovely time	29
have a	37,433
a lovely	14

The 4-gram *have a lovely time* occurs with a frequency of 15. The 3-grams *have a lovely* and *a lovely time* occur 58 and 44 times respectively. 15 of those occurrences are, however, occurrences as part of the superstring *have a lovely time*. To get the consolidated frequency of occurrence for *have a lovely* and *a lovely time* (i.e. the occurrences of these 3-grams on their own, not counting when they occur in the longer string), we therefore deduct the frequency of their superstring (15) from their own frequency as shown as in Fig. 2, step 1. This results in a consolidated frequency of 43 for *have a lovely* (i.e. $58 - 15$) and 29 for *a lovely time* (i.e. $44 - 15$). The 2-grams *have a, a lovely* and *lovely time* are also substrings of *have a lovely time* and therefore also need to have their frequency reduced by 15, resulting in a frequency of 37,476 for *have a*, 86 for *a lovely* and 29 for *lovely time* (Fig. 2, step 2). In addition, *have a* and *a lovely* are substrings of *have a lovely* and therefore the frequency of *have a lovely*, which is now 43, needs to be deducted from their frequencies (Fig. 2, step 3). This results in a new frequency of 37,433 for *have a* $(37,476 - 43)$ and 43 for *a lovely* $(86 - 43)$. *a lovely* and *lovely time* are furthermore substrings of *a lovely time* and consequently need to have their frequencies reduced by that of *a lovely time* (i.e. by 29): the consolidated frequency of *a lovely* is now 14 (i.e. $43 - 29$), that of *lovely time* is now zero. The final output of the frequency consolidation is given in Example 3. We note that *lovely time* is completely eliminated and does not appear in Example 3 since it has a consolidated frequency of zero. This type of substring reduction is an automatic consequence of the frequency consolidation. The example of the 'lovely time'-cluster also shows that the type of data provided in Example 3 can usefully complement the bare frequencies in Example 2 for purposes of a linguistic analysis. Although in this example the consolidated frequencies could easily be worked out at the time of analysis, in reality a cluster is not artificially isolated as in this example and the large net of sub- and superstrings that need to be considered makes a consolidation extremely laborious to work out manually.

The procedure as narrated above is expressed in pseudo code in Fig. 3. Iterative loops enable the processing of any (reasonable) number of different n-gram lengths, far beyond the three lengths of Fig. 2.

Assuming for a moment that Example 2 (above) represents the entire set of n-grams extractable from the underlying source text, the accuracy of results in Example 3 could be assessed by comparing the word count of the source text with the number of words bound up in Example 3, that is, the length in words of each n-gram, multiplied by its frequency and summed: $\sum(|n|^{1..n} \cdot f^{1..n})$. For Example 3, the numbers would match and confirm the accuracy of the

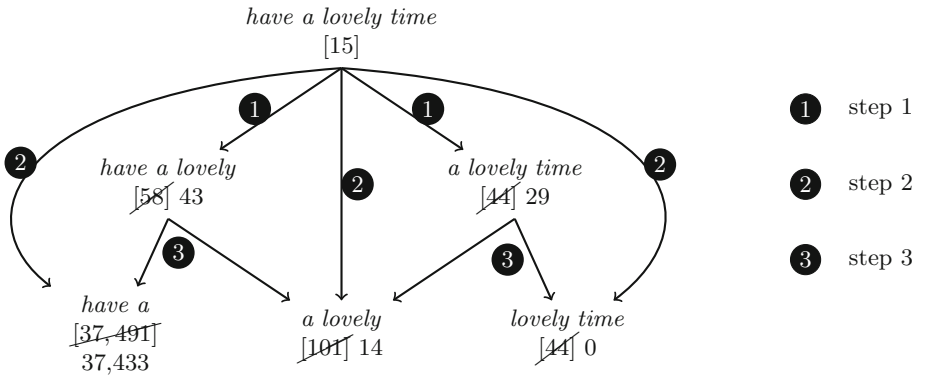

Fig. 2. Consolidation of n-grams in Example 2. Arrows indicate frequency subtractions: the frequencies at their starting points are deducted from the frequencies at their end points.

```
FUNCTION CONSOLIDATE(firstList, secondList)
   accept 2 arguments: firstList, secondList
   FOR each line IN firstList
      cut off frequency at the end of the line
      store frequency in Freq1
      store the rest in SearchLine
      search secondList for lines containing SearchLine
      IF matching lines are found THEN
         sum the n-gram frequencies of each matching line
         store the result in Freq2
         subtract Freq2 from Freq1 and store result in newFreq
         IF newFreq > 0 THEN
            replace frequency information in original line in
            firstList with newFreq
         ELSE
            delete original line from firstList
         END IF
      END IF
   END FOR
END FUNCTION

# the function is now applied to input lists
SET LongListIndex to the total number of input lists present
SET LongListMinusIndex to [LongListIndex - 1]
REPEAT
   CONSOLIDATE(List[LongListMinusIndex], List[LongListIndex])
   SET LongListIndex2 to [LongListIndex - 1]
   REPEAT
      CONSOLIDATE(List[LongListMinusIndex], List[LongListIndex2])
      SUBSTRACT 1 from LongListIndex2
   UNTIL LongListMinusIndex = LongListIndex2
   SUBTRACT 1 from LongListMinusIndex
UNTIL 1 > LongListMinusIndex
```

Fig. 3. Frequency consolidation algorithm in pseudo code. Input consists of lists of n-grams named List^1, List^2, ... List^n, where List^1 contains the shortest and List^n the longest n-grams. Lists consist of n-grams of one length, one n-gram (followed by its frequency) per line.

consolidation. Summing the word tokens bound up in Example 2, on the other hand, would result in the token inflation typical of unconsolidated n-gram lists. This simple assessment only works if the frequency consolidation procedure is applied to n-gram lists containing the complete set of extractable continuous n-grams in the source text, barring n-grams across sentence boundaries, and n-grams are extracted up to an n-gram length ($|n|$) where $|n|$ equals the number of words in the longest sentence. In applying the proposed procedure to real-life uses, where these conditions do not hold, we are faced with three challenges: (i) a reasonable maximum n-gram length must be set at which the procedure is nevertheless able to resolve overlapping n-grams; (ii) the correct functioning of the procedure must be maintained even if input n-gram lists were filtered before the application of frequency consolidation (for example by the application of frequency cut-offs, cut-offs based on statistical measures of association, or other filters designed to remove uninteresting n-grams); (iii) the question of how results can be verified when the source corpus word count can no longer be used as a target word count. The exclusion of n-grams across sentence boundaries, on the other hand, can be meaningful in many real-life contexts and is therefore retained as a precondition for the application of the procedure. The possible application of the proposed procedure to non-continuous n-grams is discussed in Sect. 4.

In dealing with the three challenges identified, we consider two more examples. Figure 4 presents the same n-gram cluster as Fig. 2 except that the 4-gram *have a lovely time* is absent. The absence of such resolving superstrings can be due to a previous application of a filter or a maximum n-gram extraction length of $|n| = 3$ (the resolving superstring in this case being of length $|n| = 4$). Considering the situation in Fig. 4, we observe that without the resolving superstring *have a lovely time* (greyed out), the consolidation fails to yield accurate results, producing a negative frequency for *a lovely*. The cause of this inaccuracy lies in the inflation of n-gram frequencies created by the unresolved overlap of *have a lovely* and *a lovely time*: there are fifteen extra, unwarranted, n-gram tokens in the system which push *a lovely* into negative frequency.

However, as shown in Fig. 5, unresolved overlaps do not necessarily cause negative frequencies. The consolidation here does not take the overlap resolving superstring *not at all certain* (in grey) into account. While the incorporation of the string in grey would lead to a somewhat different result (a result that is more faithful to the conditions pertaining in the source text), the consolidated frequencies in Fig. 5 nevertheless avoid n-gram frequency inflation (and consequently negative frequencies) and are therefore considered accurate here.

Concerning the three challenges of maximum n-gram length, filtered source lists and verification, we consequently note that, firstly, inaccuracies in consolidation manifest themselves as negative frequencies and hence the accuracy of the procedure can be assessed via instances of negative frequencies. Secondly, both filtering and maximum n-gram length determine the presence or absence of overlap-resolving superstrings and thus influence the accuracy of any particular frequency consolidation process. It is therefore important to retain the highest

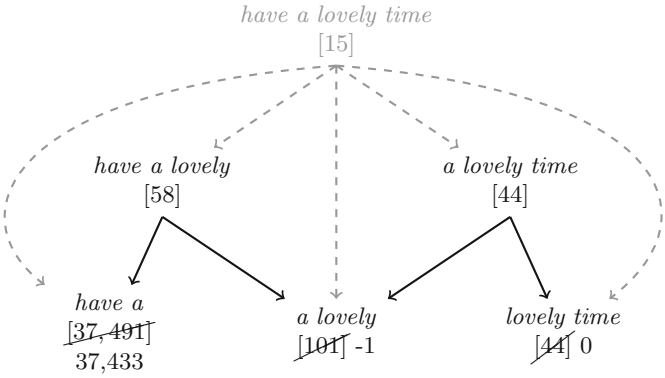

Fig. 4. Consolidation of a group of n-grams with missing resolving superstring (in grey)

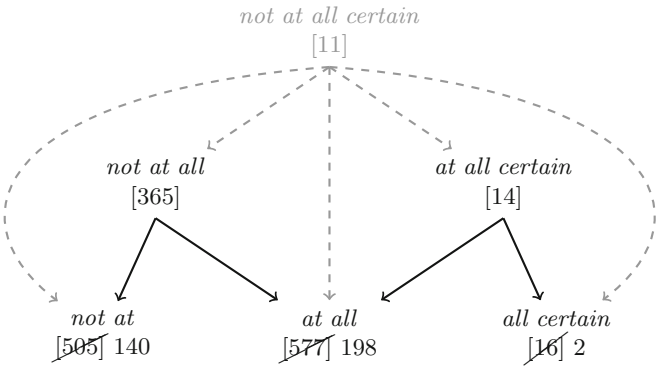

Fig. 5. Overlap without negative frequency. The overlap resolving superstring (in grey) is not considered in the consolidation shown.

number of overlap-resolving superstrings which would, if absent, cause negative frequencies. Such superstrings are hereafter referred to as *necessary superstrings*.

To safeguard the extraction of such necessary superstrings, n-grams should minimally be extracted up to length $|n| = 6$. In our test data set A (cf. Table 1 below), no missing superstrings longer than 6-grams were found to be the cause of negative frequencies. Given the extreme rarity of frequent, long n-grams, however, employing a maximum n-gram length below this length is unlikely to have a serious negative impact.

Further, the retention of necessary superstrings can be optimized by (temporarily) re-importing necessary superstrings that were eliminated by filters prior to frequency consolidation. This can be accomplished using an optional preparatory stage to the core algorithm. The preparatory stage scans filtered n-gram lists and constructs projected necessary superstrings. If these are not found among the n-grams of the filtered lists, it searches for them in the unfiltered state of the

lists (which need to be supplied) and, if found, imports them back into the filtered lists. After the frequency consolidation and substring reduction process has concluded, the imported n-grams are easily eliminated by the re-application of the relevant filter(s), if desired. To avoid a situation where the imported n-grams themselves create a need for yet more necessary superstrings, only n-grams of length $|n| > 4$ (i.e. 5-grams and longer) are made available for re-import and since extensive testing showed that no necessary superstrings had frequencies of <1 per million tokens, no superstrings below that frequency are made available for re-import as part of the procedure.

4 Evaluation and Limitations

The application of the procedure to test data using parameters shown in Table 1 produced the figures in Table 2. For comparison, figures resulting from the application of Lü et al.'s SSR-algorithm [30] as implemented in Zhang's NGramTool [40] are also included. Looking at rows one and two of Table 2, the greater number of deleted substring types and tokens in row one compared to row two is made possible by the consolidation of frequencies even if frequencies of substring and superstring are not identical. This resulted in the deletion of additional substrings that featured consolidated frequencies greater than zero but below the minimum frequency (extraction parameters as per Table 1). The procedure is therefore effective in reducing the number of redundant substrings among extracted n-grams of various lengths as well as producing consolidated frequency values for all n-grams. This in turn facilitates and assessment of MWE-density: the number of word tokens that are part of the extracted word n-grams can be calculated by multiplying frequency with length for each word n-gram and summing the resulting figures. In case of the procedure in row one, this comes to 1,748,239. Given a corpus size of four million word tokens, this is a proportion of 43.7% of running words.

The number of inaccuracies in row one as measured by the number of n-grams with negative frequencies is modest compared to the overall number of types. This number is further reduced by the application of the preparatory

Table 1. Extraction parameters for test data

	Data set A	Data set B
Size	4 million tokens	29 million tokens
Language	German	English
Source	Swiss Text Corpus [41]	Wikipedia
Extraction	2-grams to 7-grams	2-grams to 9-grams
Filters	Additive stop list of the 200 most frequent word forms of German; min. freq. 4 per million words (16)	Additive stop list of the 200 most frequent word forms of English; min. freq. 2 per million words (58)

Table 2. Figures resulting from various procedure variants

Data	Proc	Types			Tokens		
		Before	After	neg-freq	Before	After	neg-freq
A	1	21,953	19,696	84	1,085,102	788,839	−962
A	2	21,953	21,535	-	1,085,102	1,072,077	-
A	3	21,953	19,627	52	1,085,102	787,204	−644
A	4	44,297	39,607	708	2,220,699	1,455,015	−36,639
A	5	22,116	19,961	106	1,569,771	1,163,707	−6,228
B	2	63,680	60,958	-	12,702,769	12,277,406	-
B	3	63,680	45,297	4,893	12,702,769	6,663,964	−956,542

Note. Procedure (Proc): 1 = without preparatory stage; 2 = SSR according to [30]; 3 = with preparatory stage; 4 = window of size n +1; 5 = window of size n + 1, filtered to remove n-grams that do not occur at all in continuous form. Numbers featuring negative frequencies are not included in numbers after procedure.

stage which yielded the figures in row three of Table 2. Here, a total of 261 previously filtered-out n-gram types were re-imported. The frequency filter was re-applied at the end of the process.

If we consider results from data set B (last two rows of Table 2), a similar pattern to procedures 2 and 3 applied to data set A emerges. It is worth noting, however, that n-grams with negative frequencies are proportionally higher in data set B (procedure 3); roughly 7.5% of unconsolidated types and tokens are given negative frequencies. This shows that, without recourse to the underlying corpus, consolidation can only approximate the a consolidation that takes into account the full context of n-grams within the source corpus and the accuracy of the procedure varies depending on the data set. An analysis of types with negative frequencies in set A furthermore showed that these were, with very few exceptions, 2-grams and the negative frequency was caused by the absence of resolving superstrings of length $|n| = 4$, which were not made available for re-import by the preparatory stage. If n-grams of length $|n| = 4$ had been admitted for re-import, the result would have been a higher number of types with negative frequencies, since the larger number of re-imported n-grams would in turn have created a need for yet more resolving superstrings.

To capture patterns with variable slots such as those shown in Examples 4 to 6, n-grams are often extracted within a window such that for n-gram length $|n|$, the window size is $n+1$, $n+2$, $\ldots n+n$, allowing intervening words to be skipped. To test how the proposed frequency consolidation procedure fares with discontinuous n-grams, lists with a window size of $n+1$ (remaining parameters as per Table 1) were also extracted and frequency-consolidated using both the preparatory stage and the core algorithm. The result, shown in row four of Table 2, indicates a much lower accuracy of the procedure compared to continuous n-grams. However, applying a filter which only admits discontinuous n-gram types that are also attested in continuous form (such as those in Examples 4 to 6 where the variable slot is optional) yielded the figures in row five of Table 2 and

performed acceptably on our accuracy measure. The proposed procedure therefore cannot be recommended for a frequency consolidation among discontinuous n-grams, although the inclusion of discontinuous n-grams that also appear in continuous form does not pose difficulties for the procedure.

Example 4. With the [occasional / sole / possible / notable] exception of

Example 5. the [provisional] IRA

Example 6. on [false / supposed / fabricated / 11 / two / fresh] charges of

Finally, it was mentioned above that all negative frequencies indicate inaccuracies in the substring reduction process. Normally, the converse also applies (i.e. all inaccuracies are indicated by negative frequencies), but there is one caveat: where a string that would have received negative frequency (such as *a lovely* in Fig. 4) is filtered out prior to the frequency consolidation process, the inaccuracy, though present, is not flagged up. To assess the extent of this under-reporting, test data set A was run through procedure 3 with one alteration: the 2-gram list, from which the vast majority of negative-frequency types stem, had no stop list applied and all n-grams with a minimum frequency of 2 were admitted. This resulted in a 2-gram list of 244,611 types compared to 16,764 types in the procedure that produced the figures of row three. The substring reduction process produced 53 types with negative frequencies, only one type more than the 52 types previously obtained in Table 2, row three. A further test applied a more severe 2-gram stop list that produced a 2-gram list of merely 9,554 types. The number of types with negative frequencies was only moderately affected, showing a count of 33. We can conclude that all negative frequencies indicate inaccuracies and that, while all inaccuracies are not necessarily indicated by negative frequencies, the figure is very close. The number of types with negative frequencies therefore remains an excellent indicator of the accuracy of a particular substring reduction process.

5 Conclusions

A frequency consolidation procedure was presented which is able to fully consolidate frequencies of word n-grams of various lengths to a high degree of accuracy. The output of the procedure can be used to gain fast and easy access to consolidated frequencies for linguistic analysis, to calculate the proportion of text that is part of recurring word n-grams and to reduce the number of redundant substrings in a data set (substring reduction). A means of assessing the accuracy of a particular substring-reduction process was also suggested. Preconditions for the application of the suggested procedure are the non-extraction of n-grams across sentence boundaries and the extraction of n-grams up to a reasonable length (a length of 6 is minimally suggested). It was also found that the procedure works best with continuous n-grams but can be used to consolidate n-grams with optional slots. The suggested procedure has several advantages over other

existing attempts at frequency consolidation which are either inextricably linked to particular n-gram extraction procedures and therefore cannot be used where full access to source texts is unavailable or only deal with substring reduction or with the identification of an ideal n-gram length among a cluster of n-grams. The proposed procedure therefore facilitates the addressing important theoretical questions and practical challenges in the area of corpus-based MWE research, even in situations where existing other procedures are inapplicable. The procedure is implemented and available for use as part of the open source software programme SubString [1].

References

1. Buerki A.: SubString 0.9.9 (Computer Software) (2016). http://buerki.github.io/SubString/
2. Sinclair, J.: Corpus, Concordance, Collocation. Oxford University Press, Oxford (1991)
3. Hunston, S., Francis, G.: PatternGrammar: A Corpus-driven Approach to the Lexical Grammar of English. John Benjamins, Amsterdam (2000)
4. Hoey, M.: Lexical Priming: A New Theory of Words and Language. Routledge, London (2005)
5. Fillmore, C.J., Kay, P., O'Connor, M.C.: Regularity and idiomaticity in grammatical constructions: the case of let alone. Language **64**(3), 501–538 (1988)
6. Goldberg, A.E.: Constructions: A Construction Grammar Approach to Argument Structure. The University of Chicago Press, Chicago (1995)
7. Hilpert, M.: Construction Grammar and its Application to English. Edinburgh University Press, Edinburgh (2014)
8. Wray, A.: What do we (think we) know about formulaic language? An evaluation of the current state of play. Ann. Rev. Appl. Linguist. **32**, 231–254 (2012)
9. Villavicencio, A., Bond, F., Korhonen, A., McCarthy, D.: Editorial: introduction to the special issue on multiword expressions: having a crack at a hard nut. Comput. Speech Lang. **19**(4), 365–377 (2005)
10. Bouamor, D., Semmar, N., Zweigenbaum, P.: Improved statistical machine translation using multiword expressions. In: LIHMT 2011, pp. 15–20 (2011)
11. Ren, Z., Lü, Y., Cao, J., Liu, Q., Huang, Y.: Improving statistical machine translation using domain bilingual multiword expressions. In: Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications, pp. 47–54 (2009)
12. Rayson, P., Piao, S., Sharoff, S., Evert, S., Moirón, B.V.: Multiword expressions: hard going or plain sailing? Lang. Resour. Eval. **44**(1), 1–5 (2010)
13. Seretan, V.: Syntax-Based Collocation Extraction. Springer, Dordrecht (2011). doi:10.1007/978-94-007-0134-2
14. Pearce, D.: A comparative evaluation of collocation extraction techniques. In: Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002), pp. 1530–1536 (2002)
15. Sinclair, J.: Trust the Text. Routledge, London (2004)
16. Biber, D., Johansson, S., Leech, G., Conrad, S., Finegan, E.: Longman Grammar of Spoken and Written English. Pearson Education, Harlow (1999)
17. Erman, B., Warren, B.: The idiom principle and the open choice principle. Text **20**(1), 29–62 (2000)

18. Altenberg, B.: On the phraseology of spoken English: the evidence of recurrentword-combinations. In: Cowie, A.P. (ed.) Phraseology: Theory, Analysis and Applications, pp. 101–122. Clarendon Press, Oxford (1998)
19. Mittmann, B.: Mehrwort-Cluster in der englischen Alltagskonversation: Unterschiede zwischen britischem und amerikanischem gesprochenen Englisch als Indikatoren für den präfabrizierten Charakter der Sprache, G. Narr, Tübingen (2004)
20. Sorhus, H.B.: To hear ourselves - implications for teaching English as a second language. Engl. Lang. Teach. J. **31**(3), 211–221 (1977)
21. Van Lancker, S.D.: When novel sentences spoken or heard for the first time in the history of the universe are not enough: toward a dual-process model of language. Int. J. Lang. Commun. Disord. **39**(1), 1–44 (2004)
22. Altenberg, B., Eeg-Olofsson, M.: Presentation of a project. In: Aarts, J., Meijs, W. (eds.) Theory and Practice in Corpus Linguistics, pp. 1–26. Rodopi, Amsterdam (1990)
23. Smadja, F.Z.: Retrieving collocations from text: Xtract. Comput. Linguist. **19**(1), 143–177 (1994)
24. O'Donnell, M.B.: The adjusted frequency list: a method to produce cluster-sensitive frequency lists. ICAME J. **35**, 135–169 (2011)
25. Church, K., Yamamoto, M.: Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus. Comput. Linguist. **27**, 1–30 (2001)
26. Michel, J.B., Shen, Y.K., Aiden, A.P., Veres, A., Gray, M.K., Pickett, J.P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J.: Quantitative analysis of culture using millions of digitized books. Science **331**(6014), 176–182 (2011)
27. Davies, M.: N-grams data Corpus of Contemporary American English. http://www.ngrams.info. Accessed 12 June 2017
28. Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., Suchomel, V.: The sketch engine: ten years on. Lexicography **1**(1), 7–36 (2014)
29. Rayson, P.: A web-based corpus processing environment. http://ucrel.lancs.ac.uk/wmatrix/
30. Lü, X., Zhang, L., Hu, J.: Statistical substring reduction in linear time. In: Su, K.-Y., Tsujii, J., Lee, J.-H., Kwong, O.Y. (eds.) IJCNLP 2004. LNCS (LNAI), vol. 3248, pp. 320–327. Springer, Heidelberg (2005). doi:10.1007/978-3-540-30211-7_34
31. Wible, D., Tsao, N.L.: StringNet as a computational resource for discovering and investigating linguistic constructions. In: Proceedings of the NAACL HLT Workshop on Extracting and Using Constructions in Computational Linguistics, pp. 25–31 (2010)
32. Sketch Engine User Guide. https://www.sketchengine.co.uk/user-guide/user-manual/n-grams/. Accessed 10 June 2017
33. Kita, K., Kato, Y., Omoto, T., Yano, Y.: Mutual information vs. cost criteria. J. Nat. Lang. Proc. **1**(1), 21–33 (1994)
34. Frantzi, K., Ananiadou, S., Mima, H.: Automatic recognition of multi-word terms: the C-value/NC-value method. Int. J. Digit. Libr. **3**(2), 115–130 (2000)
35. Frantzi, K.T., Ananiadou, S.: Extracting nested collocations. In: Proceedings of the 16th conference on Computational linguistics, vol. 1, pp. 41–46 (1996)
36. Gries, S.T.: A bigram gravity approach to the homogeneity of corpora. In: Proceedings of Corpus Linguistics (2009)
37. Gries, S.T., Mukherjee, J.: Lexical gravity across varieties of English: an ICE-based study of n-grams in Asian Englishes. Int. J. Corpus Linguist. **15**(4), 520–548 (2010)
38. Daudaravicius, V., Marcinkeviciene, R.: Gravity counts for the boundaries of collocations. Int. J. Corpus Linguist. **9**(2), 321–348 (2004)

39. Gries, S., Wahl, A.: A new recursive approach towards multiword expression extraction and four small validation case studies. Paper Presented at Corpus Linguistics 2017, University of Birmingham, 25 July 2017
40. Zhang, L.: NGramTool (Computer Software) (2004). http://homepages.inf.ed.ac.uk/lzhang10/ngram.html
41. Bickel, H., Gasser, M., Häcki Buhofer, A., Hofer, L., Schön, C.H.: Schweizer Text Korpus - Theoretische Grundlagen, Korpusdesign und Abfragemöglichkeiten. Linguistik Online **39**(3), 5–31 (2009)