Computational extraction of formulaic sequences from corpora

Two case studies of a new extraction algorithm

Alexander Wahl and Stefan Th. Gries
Donders Institute for Brain, Cognition and Behaviour, Radboud University / University of California Santa Barbara & Justus Liebig University

We describe a new algorithm for the extraction of formulaic language from corpora. Entitled MERGE (Multi-word Expressions from the Recursive Grouping of Elements), it iteratively combines adjacent bigrams into progressively longer sequences based on lexical association strengths. We then provide empirical evidence for this approach via two case studies. First, we compare the performance of MERGE to that of another algorithm by examining the outputs of the approaches compared with manually annotated formulaic sequences from the spoken component of the British National Corpus. Second, we employ two child language corpora to examine whether MERGE can predict the formulas that the children learn based on caregiver input. Ultimately, we show that MERGE indeed performs well, offering a powerful approach for the extraction of formulas.

Keywords: formulaic sequences, collocation extraction, lexical association, child language, MERGE, adjusted frequency list

1. Introduction

Bolinger (1976, p. 2) famously claimed that "speakers do at least as much remembering as they do putting together", suggesting that the production of complex linguistic constituents (e.g. multiword phrases) was as often about retrieving these items from memory in pre-fabricated form as it was about constructing them online based on regular rules. While at the time such a view was seen as radical, today an increasing number of studies are examining the importance, complexity, and ubiquitousness of such *formulaic language* or *phraseology* (see Wray, 2002; Granger and Meunier, 2008 for influential discussion and overviews.)

https://doi.org/10.1075/ivitra.24.05wah

This term broadly encompasses many different types of multiword and co-occurrence phenomena. Prototypical or well-known kinds of formulaic language include idioms (*kick the bucket*), prepositional verbs (*talk about*), phrasal verbs (*pick up*), multiword prepositions (*in spite of*), and nominal compounds (*gold medal*), among others. But formulaicity operates at a more subtle level, too. Consider the well-known example of the two semantically similar words *strong* and *powerful*, where only the former is typically applied to the noun *tea*. This case demonstrates the phenomenon of *restricted exchangeability* (Erman and Warren, 2000), whereby formulaic language may be diagnosed when one or more words in a word sequence could not be substituted with synonyms without a loss in the particular meaning of that sequence. The implication is that the production of the noun phrase 'strong tea' cannot be based purely on a generative phrase structure rule agnostic to the lexical combinatorial preferences of individual words; rather, the language user must store some knowledge that circumscribes a complete phrasal unit populated with these particular lexical items. In other words, the whole is more than the sum of its parts.

While restricted exchangeability is of limited use in cases where there are no suitable synonyms (e.g. when a word sequence comprises only function words), Erman and Warren (2000) determined, primarily based on this criterion, that at least 50% to 60% of the corpora they examined comprised formulaic language. Numerous other studies have yielded formulaic sequence density estimates as well, with often wildly different results and, because of differences in diagnostic criteria, some counts of corpus formulaic language density going as high as 80% (Altenberg, 1998). This all suggests that Bolinger's historic claim, while hard to verify numerically exactly, may have essentially been correct. Ultimately, regardless of exact density, it is clear that formulaic language is an important feature of language that was ignored in much of mainstream linguistics until work from a phraseological perspective (e.g. Wray, 2002) and work from a usage-based perspective on how much is stored and how much is computed (e.g. Bybee, 2010) zoomed into what had largely been a computational-linguistic task/phenomenon.

In order to study formulaic language (or collocations), one must be able to identify these sequences in discourse. However, this is no straightforward task. One option would be to annotate sequences by hand, but then sequence identification criteria must be defined. Perhaps the most frequent approach is to simply ask annotators who have specialist-level familiarity with formulaic language to perform the task (e.g. Ellis et al., 2008). An obvious objection to such an approach would be the chance for bias, so annotations from different raters are often compared in order to arrive at a reasonably consistent set of annotations. Still, the nature of the mental criteria individual raters are applying is not necessarily clear.

Alternatively, annotators may be provided with more specific instructions in how to identify formulaic language, yet these are prone to the problem of formulaic

language definitions typically being insufficiently comprehensive. So, the aforementioned restricted exchangeability used by Erman and Warren (2000) is a succinct criterion and works well for certain sequences, but it cannot be applied in cases where, as mentioned above, there are no suitable synonyms to exchange for a given word in the sequence (to check whether that sequence thereby loses its idiomaticity under the exchange).

Finally, one could define more elaborate annotation criteria – for example questions aimed at identifying specific types of formulaic sequences (e.g. "is this sequence a nominal compound comprising two or more nouns with a non-compositional meaning?" or, "does this sequence function as a single multiword preposition?" etc.). Yet even still, certain obviously formulaic sequences can be difficult to definitively categorize, particularly in the case of sequences that do not co-extend with syntactic constituents (see Biber et al., 2004). In addition, as is particularly true of this last approach, manual annotation is slow and backbreaking work.

Ideally, one would want to be able to extract a reasonably reliable list of formulaic sequences from a corpus without an excessive amount of manpower. For this reason, a widely-used alternative to manual annotation is different collocational extraction algorithms, implemented computationally and applied to corpora. The algorithms vary in their designs, but they all return an ordered list of multiword sequences, whose ranking may be thought of as representing the confidence of the algorithm in the degree to which any sequence represents a true formulaic sequence. This ranking is assembled according to some statistical measure – which is itself based on the frequency of each sequence and the contingency/predictability of its parts – but the particular statistical measure used often varies from algorithm to algorithm; see below.

Thus, broadly speaking, automatic extraction is successful insofar as usage frequency is correlated with formulaicity. And indeed, much research has shown that the more often language users deploy a particular formulation rather than an alternative one with the same meaning, that formulation increasingly becomes (via statistical preemption) the conventionalized way of expressing oneself (and, in turn, comes to no longer mean "the same thing" as erstwhile alternatives) (e.g. Bybee, 2010, Chapter 3). At the same time, the results of automatic extraction algorithms are still noisy. The reason for this noisiness is twofold: On the one hand, this has to do with the fact that this correlation between usage and formulaicity is not perfect. On the other hand, different algorithms yield results that differ in their goals (e.g. lexicographic and translational goals differ) and their methodological implementation (statistical algorithms react differently to input frequencies), which affects their output and, thus, also their quality.

In the current study, we present an algorithm that we have developed entitled MERGE (for Multi-Word Expressions from the Recursive Grouping of

Elements).[1] We believe that our approach addresses some of the limitations of previous approaches from the literature, with regard to both these issues of counting sequences and identifying them. To investigate the degree to which this is true, we formulate the following research question:

RQ1: Does our algorithm perform better than a more conventional approach when both are compared to manual annotation?
Relatedly, remember that the ultimate goal for formulaic sequence identification is often some downstream research such as variety research, psycholinguistic processing, and L1 acquisition. For example, researchers have examined, among other things, dialectological differences on the basis of differences in formulaic language (Gries and Mukherjee, 2010); the degree to which formulaic sequences are processed more quickly than non-formulaic ones by adults (Arnon and Snider, 2010); and the degree to which formulaic sequences play a role in early child language (Lieven et al., 2009). And while many such approaches rely on manual annotation (but see Gries and Muhkerjee, 2010), if a particular corpus extraction approach is viable, it ought to be possible to put this to use in place of manual annotation. Thus, a second research question that we pursue is:

RQ2: Can an extraction algorithm be successfully employed as part of the methodology of a formulaic language-focused study?

In the next three subsections, we discuss in more detail the issues surrounding contemporary computational extraction approaches. Then, in Section 2, we define our extraction approach. The sections that follow comprise case studies: Section 3 evaluates our approach on the basis of annotated corpus data and aims to address the first research question, while Section 4 addresses the second research question by demonstrating the applicability of our approach to formulaic language research through a small case study on child language. Finally, in Section 5, we discuss conclusions and directions for future research.

1.1 Counting co-occurrences

One of the most important variables affecting the performance of different automatic extraction approaches is the statistic a particular algorithm uses to weight or merge word co-occurrences. Probably the two most popular methods, or families

1. We use the terms multiword expression (or MWE), formula, and formulaic sequence interchangeably here.

of methods, are (i) *relative frequencies*, which are simply the frequencies of a co-occurrence normalized, typically, for the frequency of the first/node word of a collocation/formulaic sequence, and (ii) *lexical association measures*. Numerous association measures have been proposed (e.g. Pecina (2009) reviews 80), and they vary mathematically and, therefore, in the precise list of results that they return. However, generally speaking, the most widely-used association measures are based on how much more or less often a particular sequence is observed than might be expected by chance. Such scores are calculated by considering not just the frequency of the target sequence, but other pieces of frequency information relevant to the occurrence as well. Depending on the specific measure, this may include the frequencies of the individual words (see above), as well as the size of the corpus (usually measured in words).

Most of these measures are based on *contingency tables*, such as the one in Table 1, which schematically represents the observed and expected frequencies of occurrence of the two constituents of a bigram (or any bipartite co-occurrence, for that matter).

Table 1. Schematic 2×2 table for bigram co-occurrence statistics / association measures

	word$_2$ = present	**word$_2$ = absent**	**Total**
word$_1$ = present	*obs.*: *a* *exp.*: $^{(a+b)\times(a+c)}/_{n}$	*obs.*: *b* *exp.*: $^{(a+b)\times(b+d)}/_{n}$	$a+b$
word$_1$ = absent	*obs.*: *c* *exp.*: $^{(c+d)\times(a+c)}/_{n}$	*obs.*: *d* *exp.*: $^{(c+d)\times(b+d)}/_{n}$	$c+d$
Totals	$a+c$	$b+d$	$a+b+c+d=n$

Based on the frequencies represented in Table 1, an association measure returns an association score for each co-occurrence type; these scores may then be used to rank the bigrams in a corpus by strength or significance. While each measure's scores represent different units, often a positive value will indicate statistical attraction between two words: that is, that the two words co-occur more often than might be expected by chance. Conversely, a negative value will indicate statistical repulsion, or that two words occur less frequently than might be expected by chance (see Evert, 2004, 2009 for comprehensive discussion).

Lexical association measures tend to offer greater sensitivity to formulaic language than relative frequency, since they can capture sequences that are infrequent though nonetheless formulaic. Consider the bigrams *San Francisco* and *in the*. While the latter sequence is clearly more frequent (and would thus be ranked more highly on a frequency list), most would agree that the former is a 'better' formulaic sequence. This is because when one of the unigrams *San* and *Francisco* does occur, there is a high probability that the other will, too, whereas when *in* and *the* occur, they may occur together but they very often occur apart as well. In other words,

San and *Francisco* embody a much greater degree of contingency than do *in* and *the*. It is this feature that most lexical association measures are designed to capture.

Of the measures that have been developed, some have emerged as more popular than others. For example, pointwise mutual information (MI) is probably the most well-known association measure. However, MI and transitional probability – which is not usually considered a lexical association measure but nonetheless measures sequence strength – exhibit a similar problem: they often rank very low-frequency, high-contingency bigrams too highly, even in the case of a bigram in which both component words are hapaxes (see Daudaravičius and Murcinkevičienė, 2004, pp. 325–326). In other words, these two measures have the opposite problem of relative frequency. Ideally, one would want a measure that 'splits the difference' between these two extremes. While alternatives such as MI^k fare somewhat better in this respect (see McEnery, 2006; Evert, 2009, p. 1225), one lexical association measure that has yielded quite good results for multiword extraction (e.g. Wahl, 2015), and does not appear oversensitive to very low frequencies is log-likelihood (Dunning, 1993), whose formula is given in (1).

(1) $$log\ likelihood = 2\Sigma_{i=a}^{d} obs \times log \frac{obs}{exp}$$

Unlike some other measures, log-likelihood takes into account observed and expected values from all four frequency cells (*a*, *b*, *c*, and *d*) of the kind of contingency table shown in Table 1. Because of the very widespread, successful adoption of log-likelihood in many studies (collocation studies, multiword extraction studies, keywords studies, etc.), log-likelihood is the measure we use in the algorithm that we develop here.[2]

The reader may note that we have not discussed in this section co-occurrences of higher-order *n*-grams. This is not an omission, but rather reflects the fact that virtually all lexical association measures are designed for two-way co-occurrences. This is of course problematic, since formulaic sequences may theoretically comprise any number of words. Some techniques for adapting lexical association to higher-order *n*-grams have been developed (see also below), but no best practice has emerged yet. In addition, while relative frequency does exhibit the insensitivity to low-frequency, high-contingency sequences as discussed above, it does not have the bigram restriction, and thus still is used by researchers today (e.g. O'Donnell, 2011). We return to these issues in a little while.

2. One final point that should be made is that (1) will always result in positive values. Thus, in order for log-likelihood scores to correspond to the convention in which positive values denote statistical attraction between words and negative values repulsion, the product of Equation 1 must be multiplied by –1 when the observed frequency of a bigram is less than the expected (following Evert, 2009, p. 1227).

1.2 N-Gram sizes/configurations and the problem of redundancy

Once a scoring metric has been chosen, a typical next step is to select one or more *n*-gram sizes for extraction. Furthermore, one may choose *n*-gram templates that contain one or more gaps in them. This reflects the possibility of discontinuous formulaic sequences, exemplified by the *as _ as* construction in *as tall as* or *as little as*. Next, all *n*-grams corresponding to the selected templates are extracted from a corpus, they are scored, and then they are ranked: ultimately, the higher-ranked *n*-grams are the algorithm's best hypotheses for true formulaic sequences.

However, even if one uses relative frequency for scoring or if one manages to adapt lexical association measures to co-occurrences greater than 2-grams, one still faces an issue of redundancy with this conventional approach. Specifically, if one extracts the 5-gram *as a matter of fact*, one will have also extracted the 4-grams *as a matter of* and *a matter of fact*. Because these 4-grams are at least as frequent as the 5-gram that contains them, they might be ranked higher (if ranking is based on frequency, or, in the case of a lexical association-based ranking, since strength is correlated with frequency). Of course, this effect is a problem since, in the case of *as a matter of fact*, the 5-gram is clearly a better hypothesis for a 'true' formulaic sequence than any n<5-grams included in *as a matter of fact*.

1.3 Recent approaches

Some recent approaches address the above-mentioned issues. For example, Daudaravičius and Marcinkevičienė (2004) develop a new lexical association measure called lexical gravity *G*. This measure computes the lexical association of two elements *x* and *y* by not only using the information in Table 1 above (i.e. the token frequencies with which *x* and *y* are observed in the corpus together and on their own), but also using the numbers of types with which *x* and *y* co-occur (i.e. the type frequencies underlying the token frequencies of cells *b* and *c* in Table 1). They then apply this measure to the identification of formulaic language by, so to speak, moving through a corpus incrementally and considering any uninterrupted sequence of bigrams with a *G*-score exceeding a threshold as constituting a formulaic sequence, or 'collocational chain' in their terminology.

In a later paper, Gries and Mukherjee (2010) develop a modification of lexical gravity for the identification of formulaic language. Specifically, they extract sequences of various lengths and score them on the basis of the *G*-score of their component bigrams, discarding those sequences with mean *G*-scores below a certain threshold. Then, they proceed through the list, discarding sequences that are contained by one or more n+1-grams with a higher mean *G*-score. The resulting list

constitutes their algorithm's hypothesis of the formulaic sequences in the corpus. With this pruning process, Gries and Mukherjee's approach also addresses the redundancy issue mentioned in the previous section, whereby high-scoring grams may merely be fragments of larger, true grams. However, the fact that lower-order *n*-grams are entirely discarded if a higher-order *n*-gram containing them is stronger is potentially problematic: while certain tokens of a lower-order *n*-gram may be fragmentary (*fingers crossed* in *to keep one's fingers crossed*), others may not be (*fingers crossed* in *Speaker A: "I hope we win!" Speaker B: "Fingers crossed!"*).

A recent approach by O'Donnell (2011) takes a different approach to extracting formulaic sequences of various sizes, which also avoids the problem of redundancy: Rather than adapting lexical association measures to co-occurrences beyond the bigram, O'Donnell employs frequency counts as a metric of formula strength. His Adjusted Frequency List (AFL) works by first identifying all *n*-grams up to some size threshold in a corpus. Next, only *n*-grams exceeding some frequency threshold (3, in his study) are retained in the AFL along with their frequency. Then, for each *n*-gram, starting with those of threshold length and descending by order of length, the two components *n*-minus-1-grams are derived. Finally, the number of tokens in the frequency list of each *n*-minus-1-gram is decremented by the number of *n*-grams in which it is a component. Like the approaches above, this procedure prevents the kinds of overlaps and redundancies that would result from a brute-force approach of simply extracting all *n*-grams of various sizes and then ranking them based on frequency. However, in using the AFL, there is a very real risk that low-frequency though high-contingency formulaic sequences would be ranked (too?) low, while high-frequency though non-formulaic sequences would be ranked (too?) high.

One drawback shared by all of the approaches discussed thus far is that, as implemented, they do not allow for discontinuous formulaic sequences. A recent algorithm by Wible et al. (2006) addresses this limitation. Their approach also crucially differs from these other approaches in that it does not generate a list of ranked formula hypotheses contained in a corpus. Instead, it is designed to find all of the formulaic sequences that a given node word participates in (in this way, it is more akin to a concordance). Their algorithm represents what we will call a recursive bigram approach. Upon selection of a node word to be searched, the algorithm generates continuous and discontinuous bigrams within a specified window size around each token of the node word in the corpus; these bigrams consist of all those that have the node word as one of their elements. Next, the algorithm scores the bigrams on the basis of a lexical association measure (they use MI), and all those bigrams whose score exceeds a specified threshold are 'merged' into a single representation. The algorithm then considers new continuous and discontinuous bigrams, in which one of the elements is one of the new, merged representations

and the other element is a single word within the window. The new bigrams are scored, and winners are chosen and merged. This process iterates until no more bigrams exceeding the threshold are found. Ultimately, the algorithm generates a list of formulaic sequence of various sizes that contain the original node word.

Importantly, the model never has to calculate association strengths for co-occurrences larger than two elements, since one element will always be a word, and, after the first iteration, the other element will always be a word sequence containing the node word. The obvious limitation of this approach is that it is not designed for broad-scale use on all words in a corpus. In principle, one could treat every corpus word type as a node word. However, this would result in numerous instances of redundancy, whereby partially or fully overlapping formulaic sequences would be grown from neighboring node words. And because the authors did not intend for their algorithm to be used for applications other than concordance, they do not offer a suggestion for how this might be addressed.

In the next section, we present our algorithm, which addresses all of the issues raised so far: scalability of lexical association, redundancy, discontinuity, and broad-scale use on all words in a corpus.

2. The MERGE algorithm

Similar to the algorithm developed by Wible et al. (2006), the MERGE algorithm embodies a recursive bigram approach. But unlike their work, our algorithm is designed to extract *all* formulaic sequences in a corpus – not just those that contain a particular node word. It begins by extracting all bigram tokens in the corpus. These include adjacent bigrams, and potentially bigrams with one or more words intervening, up to some user-defined discontinuity parameter (similar to Wible et al.'s use of a window). The tokens for each bigram type are counted, as are the tokens for each individual word type, and the total corpus size (in words) is tallied. Next, these values are used to calculate log-likelihood scores. The highest-scoring bigram is selected as the winner, and it is merged into a single representation; that is, it is assigned a data structure representation equivalent to the representations of individual words (this differs from Wible and colleagues' approach, wherein multiple winners were chosen at an iteration on the basis of a threshold association value). We call these representations *lexemes*. At the next stage, all tokens of co-occurring word lexemes in the corpus that instantiate the winning bigram are replaced by instances of the new, merged representation. This process by which smaller tokens are consumed by larger winners avoids the kinds of redundancy issues raised above, in which a particular word token or sequences of tokens may simultaneously participate in numerous fragmentary grams.

Frequency information and bigram statistics must then be updated. New candidate bigrams are created through the co-occurrence in the corpus of individual word lexemes with tokens of the new merged lexeme. Furthermore, certain existing candidate bigrams may have lost tokens. That is, some of these tokens may have partially overlapped with tokens of the winning bigram (i.e. they shared a particular word token). Since these word tokens in effect no longer exist, these candidates' frequency counts must be adjusted downward. Moreover, the frequency information for the individual word types found in the winner must be reduced by the number of winning bigram tokens. Finally, the corpus frequency has decreased, since individual words have been consumed by two-word sequences. After these adjustments in frequency information have been made, new bigram strengths can be calculated.

The cycle then iteratively repeats from the point at which a winning bigram is chosen above, and iterations continue until the association strength of the winning bigram reaches some user-defined minimum cut-off threshold or until a user-defined number of iterations has been completed. The output of the algorithm is a corpus, parsed in terms of formulaic sequences, and a list of lexemes, from individual words to formulaic sequences of different sizes.

Because the input to candidate bigrams at later iterations may be output from previous iterations, MERGE can grow formulaic sequences unrestricted in size (even while never considering co-occurrences larger than two items), which is similar to the Wible et al. (2006) algorithm. Another key difference, however, is that one element of their candidate bigrams must always be a single word and the other a word sequence (at least after the first iteration, where both elements are single words). In contrast, at later iterations, MERGE can choose a winning bigram that comprises two single words, a single word and a word sequence, or two word sequences. Moreover, assuming a sufficiently sized gap parameter, one element may in principal occur inside the gap of another element. Even more unusual scenarios are possible: *as _ matter* and *a _ of fact* could be interleaved to form *as a matter of fact*. Thus, there are many possible paths of successive merges that result in particular formulaic sequences, provided that the leftmost word of the two elements of a bigram never exceed the discontinuity parameter.

3. Case study 1: MERGE vs. AFL

In this case study, we address our first research question, "does our algorithm perform better than a more conventional approach when both are compared with respect to manual annotations?" To answer this question, we chose to compare the performance of MERGE to that of one of the other algorithms discussed earlier, the Adjusted Frequency List (AFL), by O'Donnell (2011).

Like MERGE, the AFL addresses the redundancy/overlap problem faced by algorithms that simply extract and rank all *n*-grams of various sizes. However, unlike MERGE, the AFL uses frequency rather than lexical association. In another study (Wahl and Gries, 2018), we show that the use of frequency reduces the quality of the formulaic sequences found by the AFL significantly, compared with those found by MERGE. However, in that study, we evaluated the performance of the two algorithms on the basis of a rating experiment conducted using naïve participants (i.e. participants who had no explicit knowledge of formulaic language and received instructions/examples describing it on the spot).

Here, we wish to see if the superior performance of MERGE holds up in a different test situation, namely a corpus already annotated for formulaic sequences. In other words, while in the previous study we assessed the performance via naïve intuitions, here we are testing performance via specialist knowledge, as those who annotated the corpus must have had some relevant lexicographic training to do so.

3.1 Materials

The corpus we use is the spoken component of the British National Corpus (BNC), which comprises approximately 10 million words. Crucially, this component of the corpus was tagged for formulaic sequences; in total, there are 436 sequence types (once tagged and all identical sequences conflated). However, a number of these sequences contain disfluencies such as *er* or *erm*. There are a total of 48 such items, and all of their 'clean' forms are also found amongst attested among the BNC's formulaic sequences. Thus, when they are removed from the list, there are only 388 total BNC items. Having worked extensively with formulaic sequences, we must point out that this estimate likely seriously underestimates the number of formulaic sequences actually present in the BNC spoken component. Consider, for example, the work of Erman and Warren (2000), who found over 50% of their corpus comprised formulaic sequences. This would mean that over 5 million words of the BNC spoken component would be distributed among a mere 388 types, which is obviously not the case – rather, the BNC annotators must have used much more conservative criteria in determining formulaic sequences than did Erman and Warren.

In order to compare the performance of the algorithms, all 388 sequence types were first obtained from the corpus. The corpus was then preprocessed so that only word strings along with utterance boundaries were retained. Next, MERGE was run for 10,000 iterations on the corpus, with the maximum gap size set to 0 (only adjacent sequences were permitted). Additionally, the AFL was run on the corpus and the top 10,000 most frequent items were selected from the list that was

generated. Note that items 9977 through 10539 in the AFL output were all tied with a frequency of 35. In order to arrive at an even 10,000 items, we randomly selected 10,000–9977 = 23 items from these 10539–9977 = 562 total tied items.[3]

The 10,000 items from each of these runs of the respective algorithms then served as the basis for comparison with respect to the 388 tagged types from the BNC.

3.2 Results

First, we checked how many of the 388 formulaic sequences from the BNC spoken were identified by the top 10000 MERGE items and by the 10000 AFL items: MERGE found 112 of the 388 formulas whereas the AFL found only 93 of the same 388 formulaic sequences. According to a one-tailed binomial test, MERGE finds a significantly higher number of formulaic sequences [binom.test(112, 388, 93/388, alternative="greater"), $p_{\text{one-tailed}} = 0.01522$); conversely, according to a second one-tailed binomial test, the AFL performs significantly worse than MERGE [binom.test(93, 388, 112/388, alternative="greater"), $p_{\text{one-tailed}} = 0.01779$].[4]

In order to more closely analyze the differing performance of MERGE and the AFL, we present Table 2, in which each column corresponds to a different category of (non-)overlap between the algorithm outputs. Thus, column A contains those items in the BNC identified by both algorithms; column B contains those identified by MERGE but not the AFL; column C contains those identified by the AFL but not by MERGE; and column D contains those BNC items identified by neither algorithm. Note that columns A and D contain only a sampling of the total number of items in those categories.

One way to explore these sets of items quantitatively is via the parameters that matter to, or are inherent to, formulaicity: frequency of occurrence, dispersion, and lexical association. Dispersion refers to how evenly tokens of a particular type are distributed in a corpus and we are using the "DP" measure of dispersion (Gries, 2008). If tokens are perfectly evenly distributed in a corpus, DP will approach 0,

---

3. In order to rule out an effect of which 23 formulas with the AFL frequency of 35 were sampled, we conducted a Monte Carlo simulation with 1,000 iterations in which the 23 formulas were replaced with 23 randomly-sampled items from all formulas with the AFL frequency of 35. The mean and 95%-confidence interval of how many of the 388 BNC-grams the AFL found were 93.03 ([93.02, 93.04], see below), which means our randomly chosen items did not skew the results in any direction (let alone in our favour).

4. We performed one-tailed tests because our first comparison of MERGE and AFL (Wahl and Gries, 2018) showed that MERGE outperformed AFL; however, even two-tailed tests proved significant for MERGE outperforming the AFL ($p_{\text{one-tailed}} = 0.02761$) and the AFL performing worse than MERGE ($p_{\text{one-tailed}} = 0.03326$).

Table 2. Comparison of attestation of BNC items among the results of the two algorithms

Column A: +M,+A (83 types)	Column B:+M,-A (29 types)	Column C: -M,+A (10 types)	Column D: -M,-A (266 types)
by way of, subject to, as usual, in case, even if, and so on, in relation to, a little, that is, next to, off of, for good, for instance, just about, for the time being, as regards, even though, each other, as it were, at once, sort of, by now, old fashioned, from time to time, of course, all round, as to, no longer, for example, kind of, in between, rather than, as opposed to, …	*in addition, whether or not, vice versa, up to date, in order, half way, depending on, up front, up until, all of a sudden, anything but, grand prix, status quo, as if, know how, per cent, in common, fed up, so as, every so often, in accordance with, as though, en suite, a great deal, less than, per annum, an awful lot, sinn fein, out of date*	*given that, in respect of, as yet, in full, for certain, in the main, near to, no matter what, with regard to, except for*	*relative to, hard up, poco a poco, now that, teeny weeny, al fresco, at large, au fait, a la, in search of, no matter how, grand mal, a la carte, as between, as from, au revoir, nom de plume, from now on, ad hominem, in return for, in place of, insofar as, as for, except for, in relation to, once more, all at once, au pairs, pate de foie gras, in vain, in proportion to, de facto, raison d'être, …*

whereas if tokens are extremely clumpily distributed (i.e. largely or even exclusively concentrated in one part of the corpus, then DP will approach 1). As a lexical association measure, we are using the MI2 measure, a version of MI that rewards *n*-grams with higher observed frequencies – log (obs a^2/exp a) – and we computed the expected frequencies on the basis of the assumption of complete independence.[5]

In order to visualize the distributional properties of the tokens in columns A-D with respect to frequency, dispersion, and MI2, Figure 1 displays empirical cumulative distribution (ECD) plots for frequency, dispersion, and lexical association respectively for all columns A-D, but our discussion will focus on the comparison of B versus C and the comparisons of both B and C with A.

At this stage of exploration of MERGE (vs. the competing algorithm), we are not yet in a position to state specific alternative hypotheses – let alone directional ones or specific effect sizes – regarding how MERGE and the AFL differ along these three parameters other than the maybe most obvious one that MERGE should behave differently with regard to MI2 because it is an algorithm whose computations involve a measure of association strength. Thus, we are restricting our discussion here to an exploratory description. With regard to the frequencies, it is obvious that

5. That means, expected frequencies were computed as they would in chi-squared tests of independence; for a 3-gram that would be $(f_{word1} \times f_{word2} \times f_{word3}) \div \text{corpus size}^2$; see Gries (2015, Section 2.2.1 for an example and why this can only be a first heuristic).

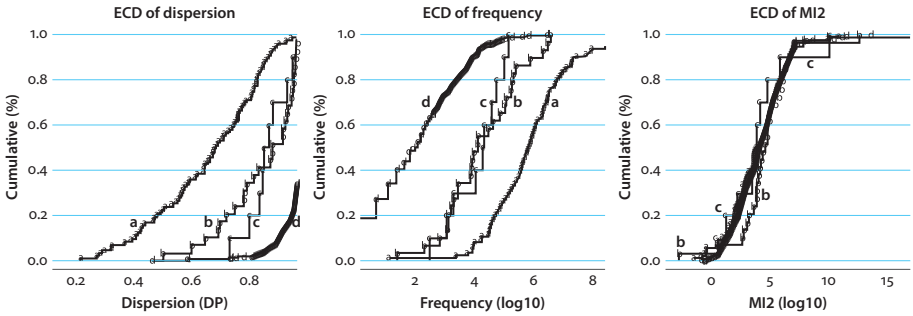

Figure 1. ECD plot of frequency (left panel), dispersion (DP, center panel), and lexical association (MI2, right panel)

(i) the formulas identified by both MERGE and the AFL are those with the highest frequencies and (ii) the formulas identified by neither MERGE nor the AFL are those with the lowest. Also, (iii) the formulas of columns B and C do not seem to differ from each other in terms of their average frequency or the variability of their frequencies, while both B and C differ from those of A (i.e. those formulas that both algorithms found). Put differently, both MERGE and the AFL agree on many high frequency collocates but the formulaic sequences that only one of them finds do not differ from each other in terms of their corpus frequencies.

With regard to dispersion, the picture changes a bit: Again, (i) the formulas identified by both algorithms are the ones with the lowest DP-values (i.e. most evenly distributed in the corpus), but it is also worth noting that the formulas found by both algorithms exhibit DP-values across the whole range of values. Then, (ii) the formulaic sequences identified by neither MERGE nor the AFL are those with the highest DP-values / clumpiness and very little variability of dispersion: 75% of the DP-values of column D are ≈ 0.96 or higher. However, (iii) while the formulaic sequences found by only one algorithm do not differ in their average dispersion, they appear to differ in the variability of their dispersion: the interquartile range of the B formulas is twice as high as that of the C formulas, which we interpret as advantageous for MERGE, because it can be seen as indicating that MERGE is better at finding formulaic sequences with diverse dispersions.

Finally, with regard to lexical association, the results are quite different: (i) the main findings are that the formulas found by at least one algorithm (i.e. those in columns A, B, and C) do not differ much from each other in terms of either central tendency or variability (with just a small effect of column A exhibiting a wider range of MI2-values). In addition, the formulas of column B do exhibit somewhat larger mean and median MI2s than those of column C, but the effect is merely suggestive at this point (in part because of the very small sample size of 10 items in column C).

Given the just-mentioned small number of cases in C, it is difficult to make a detailed qualitative comparison at this point, but it does seem to us that three of the ten column C formulas are not 'as good' examples of formulas as all of those in column B, to the extent that they seem to be incomplete or less frequent – specifically, *in respect of*, *in the main*, and *near to* – but this assessment awaits future (rating?) studies to be put on a more solid footing.

3.3 Interim conclusions

In the comparison of MERGE with the AFL in Wahl and Gries (forthcoming), we essentially employed what one might consider a kind of unsupervised approach: we ran both algorithms and then compared samples of top-ranked formulaic sequences. We found that there was a striking difference between the kinds of sequences identified by MERGE and the AFL, which patterned like the *San Francisco*/*in the* example discussed above. The present study, by contrasts, is essentially more similar to a supervised classification approach: we had a list of 388 likely positives and then the degrees to which the algorithms find them. Accordingly, we do not find the same *San Francisco*/*in the* bifurcation in the results. Rather, the results were more nuanced, with numerous items identified by both algorithms, and subtle differences in the items that were identified only by one or the other algorithm; it seems that MERGE does better in particular by being able to find formulas from a wider range of dispersion values, as well as exhibiting the tendency of identifying formulas with higher association scores.

4. Case study 2: Exploring MERGE in the context of L1 acquisition

As mentioned above, formulaic language extraction from corpora is typically a means to some other research end, used in fields as diverse as cognitive-/psycholinguistics, dialectology, digital humanities, applied linguistics, and many others. Thus, to provide evidence that an automatic extraction approach such as MERGE is powerful enough to be methodologically applicable to such downstream formulaicity research, we deploy it here in a small applied study.

Within the cognitive domain, formulaic sequences play a particularly integral role in child language. Specifically, current theories hold that they serve as a stepping stone on a child's way to more productive grammatical knowledge: children begin with stored formulaic sequences and, over time, generalize across them to acquire a mature grammar (see Tomasello, 2005 for one of the most thorough overviews); at the same time, this is not to say that representations of formulas acquired during

childhood do not endure into adulthood, nor that new formulas are not acquired beyond childhood. One question, though, is whether these early representations are truly formulaic, and not creatively constructed. One source of evidence for this would be if demonstrably formulaic structures in the adult input to the child are taken up and deployed in the child's own productions. Meanwhile, adult creative structures ought not to be reproduced by the child, at least at the same rate.

This broad style of approach, in which specific child productions are linked to specific adult inputs, has been used elsewhere in the child language literature. For example, Bod (2009) developed a parsing/grammar induction algorithm called UDOP (Unsupervised Data-Oriented Parsing). UDOP is based on a Probabilistic Context-Free Grammar (PCFG) that can store and reuse (sub)trees (including specific word terminals) that it had constructed to parse previously-encountered sentences. The lexicalized, reusable nature of the (sub)trees makes them, by definition, formulaic sequences, at least in the context of the model (whether or not they reflect true formulaic sequences known to humans is another question). The primary goal of UDOP is to demonstrate that grammatical knowledge can be induced in a bottom-up fashion, without reliance on innately-specified syntactic knowledge, contra many generative grammarians. Thus, the role of formulaic language in this model is to increase performance in the pursuit of this objective (just as a child may use formulaic language as a stepping stone).

Bod (2009) evaluated UDOP in various case studies. In one, he partitioned a longitudinal child language corpus into two sections, and then trained UDOP on the adult utterances in the earlier partition (in separate trials, he also trained the algorithm on the child utterances, and on a combination of the child and adult utterances).[6] Next, he evaluated the algorithm by seeing how well it could parse the child utterances in the later partition, based on the grammar it had acquired on the earlier adult utterances. The parses assigned were compared against manually annotated, gold standard parses for the data. Indeed, the grammar acquired based on the adult input performed well, demonstrating that a child's emergent grammatical knowledge can be modeled on concrete adult structures that the child has stored.

In another related study, Swingley (2005) examined the distributional learning of word boundaries from syllable co-occurrences. Although he did not investigate

---

6. A related approach is taken in Bannard et al.'s (2009) study using a Bayesian-based distributional learning algorithm that the authors had developed, as well as in Lieven et al.'s (2009) corpus-based discourse-analytic study. However, a crucial difference is that these studies use child utterances for both training and test; thus, there is no attempt to link the children's acquired structures to adult input, but rather just to account for the children's advancing linguistic development across different stages of the child's own usage.

formulaic word sequences, his design is instructive. He extracted all syllable bigrams and trigrams, scored them on the basis of MI and frequency, and ranked them. He then correlated this ranked list with how well the *n*-grams instantiated words. In other words, he examined the question of how well association strength and frequency can predict the word boundaries that children go on to learn. However, his definition of what children 'go on to learn' is mature, adult-like gold standard boundaries. Furthermore, the corpus he used was not longitudinal, but rather a collection of caregiver utterances (phonologically transcribed) from the input to a collection of different children. An interesting complementary approach would be to examine how well the ranked *n*-grams of (a) specific caregiver(s) predict the word boundaries that their child goes on to learn at the particular developmental stage of the corpus (which would be possible with a longitudinal corpus).

In the current chapter, our approach brings together techniques developed in evaluation methods from the child language studies within Bod (2009) and Swingley (2005). As in both approaches, we train the algorithm (MERGE) on a set of adult utterances. Like Bod (2009) and unlike Swingley (2005), we use longitudinal corpora, focusing on the input to/output from individual children. We compare the multiword representations generated by the model based on earlier adult utterances against the actual output of these children, as registered in later child utterances. And like Swingley and unlike Bod, we work with a list of output candidates scored and ranked on the basis of association strength, rather than best grammatical parses for whole utterances. The hypothesis is that higher-scoring formulaic sequences, extracted from the adult utterances, will go on to be learned/used by the child, while formulas that scored lower will not (at least not to the same degree).

In the next section, we discuss the corpora that we use as well as their pre-processing, and we discuss the technique for generating the stimulus items from the corpora using MERGE. After that, we turn to the results of the study. Finally, we discuss these findings.

4.1 Materials and methods

In this study, we use two longitudinal child language corpora, both of which were sourced from the CHILDES database (MacWhinney, 2000). CHILDES is an online repository for corpora of child language acquisition data. Our selected corpora are the 'Lara' corpus (Rowland and Fletcher, 2006) and the 'Thomas' corpus (Lieven et al., 2009). Both Lara and Thomas are children who have grown up in the United Kingdom (and were thus raised as native speakers of varieties of British English), and the recordings were made in the children's respective homes.

These corpora were selected for several reasons. First, they both span the early multiword speech stage of development, an ideal stage for examining the role of formulaic language in early acquisition: Lara was between the ages of 1;9.13 (i.e., 1 year, 9 months, and 13 days) and 3;3.25 when her recordings were made, and Thomas was between the ages of 2;00.12 and 4;11.20 when his were made. Second, both corpora include extensive speech from the children as well as caregivers with whom they interact (and, in the case of the 'Thomas' corpus, researcher speech as well). Finally, the corpora are relatively large/dense: while 'Lara' comprises 120 hours of transcribed audio, 'Thomas' totals 379 hours of transcribed audio.

The 'Thomas' recordings/transcriptions are in fact divided into 3 subcorpora. The first subcorpus spans the ages of 2;00.12 to 3;02.12, and recordings were made for 1 hour, 4 times per week. The second and third subcorpora span the remainder of the time, and recordings were made for 1 hour, once per week.[7] Because the first subcorpus overlaps in time most closely with the 'Lara' corpus, we only used those recordings. Even with this limitation, the first subcorpus still comprises 279 hours' worth of transcripts (i.e. more than double the size of the 'Lara' corpus). In order to make the corpora more comparable in size, the first 'Thomas' subcorpus was downsampled by including only every other corpus file. This resulted in a more comparable 140 hours' worth of transcripts.

Both corpora were transcribed according to the CHAT format (MacWhinney, 2000), so the same preprocessing procedure was used. This included the removal of metadata, transcriber commentary, punctuation, time stamps, non-speech vocalizations, and incomprehensible syllables. In addition, transcription tags were removed, which marked phenomena such as missing words, grammatically correct forms when an incorrect form appeared, and invented forms, among other things. Note that, while incomprehensible forms were removed, grammatically/phonologically incorrect and invented forms were themselves indeed included. Speaker tags were also removed, but not before they were used to separate each corpus into child and caregiver/adult utterances. Additionally, the two corpora were divided into two partitions, whereby the first two-thirds of each corpus represented partition A and the final third represented partition B.

MERGE was then run on the adult utterances of partition A, once for each corpus. No gaps in the formulaic sequences acquired were permitted, and the algorithm was allowed to run until the log-likelihood score of the top-scoring merge candidate reached 0 (Remember that positive log-likelihood values signify statistical attraction between bigram elements while negative values signify statistical repulsion. By this standard, all bigrams exhibiting a positive log-likelihood score

7. The 'Thomas' corpus additionally included video data, but this was not used in the present study.

are in theory formulaic sequences.). From the final output, all sequences of length 2 through 5 were retained.

Next, all *n*-grams from lengths 2 through 5 were extracted from the child utterances in partition B. From this group, any *n*-grams which also appeared among the child utterances in partition A were discarded in order to ensure that the group comprised only *n*-grams that were new attestations in the child's speech. Finally, the sequences from the MERGE output were compared to the *n*-grams from the partition B child utterances, and two lists were created. The first list comprised those MERGE output sequences that also appeared as *n*-grams in the child utterances. These are formulas that the child plausibly went on to learn in partition B from the input they received from the adult utterances in partition A. The second list comprised those MERGE output sequences that did not appear as *n*-grams in the child utterances. These are items that, despite being MERGE output from the adult utterances from partition A, did not later go on to be learned by the child. The hypothesis is that the log-likelihood scores on the basis of which the sequences were merged ought to be higher for the first 'learned' group than for the second 'nonlearned' group; this is because formulaic sequences with higher degrees of attraction are more likely candidates for acquisition by the child.

Finally, for each child, all of the sequences were grouped into numbered bins on the basis of log-likelihood scores – the lowest-numbered bin contained the sequences with the lowest scores and the highest-numbered bin the highest scores. Then, for each bin, the proportion of sequences that were learned by the child was calculated. In the eventual statistical model (discussed below), the proportions of sequences learned serve as the dependent variable (the variable being predicted). In contrast, the numbers of the bins (BIN) serve as (one of) the independent variables (the variable predicting). In other words, we are trying to predict the proportion of sequences learned by the children on the basis of the BIN, which is a proxy for the log-likelihood score/MERGE order.

Note that, since the 'Thomas' corpus is larger than that of Lara, the number of sequences extracted by MERGE is larger. As a result, we created many more bins for the 'Thomas' sequence scores (213 versus Lara's 75). This is because we wanted there to roughly be the same number of scores in each bin across children (99 scores per bin for Lara and 97 scores per bin for Thomas).[8] Also note that any particular score was placed into its bin only once; that is, if MERGE extracted two different sequences on the basis of the same score, this score would not be duplicated within the appropriate bin.

8. The final bins (i.e. the one corresponding to the highest log-likelihood scores), have slightly less than 99 and 97 items in them. Despite this, this set of bin counts and number of items per bin was chosen to ensure that the final bins came as close to possible to the other bins in terms of number of items contained.

4.2 Results

The proportions of sequences learned are plotted against the normalized bin numbers in Figure 2 for Lara (left panel) and for Thomas (right panel); bin numbers were normalized to a 0–1 range to make the values of the two children comparable. Note the consistent pattern across the two. On the right half of each plot, as one moves from mid-range log-likelihood scores to high log-likelihood scores, there is an increase in the proportion of sequences per bin that are learned by each child, which is precisely what we predicted. However, the plots also display something unexpected: Moving from the low log-likelihood scores on the left of the plots, to the mid-range scores, there is a *decrease* in the proportion of learned sequences. This pattern goes against intuition – why might this be?

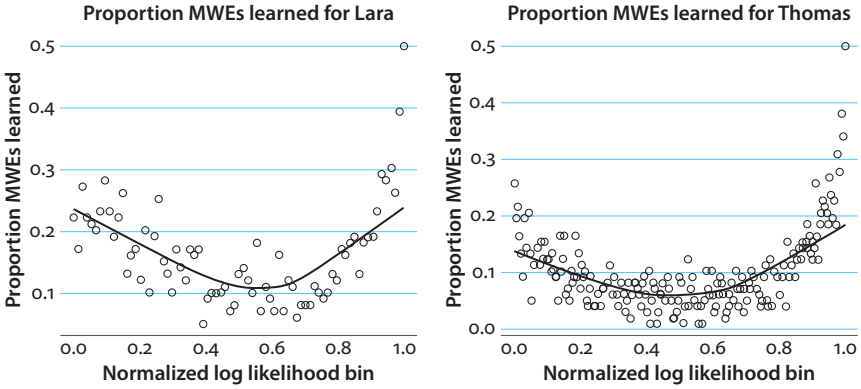

Figure 2. Proportion sequences learned as a function of normalized bin rank

One possible explanation is based on the lengths of the sequences, a factor that plays an important role in which sequences are and are not retained. Thus, in Figure 3, we show average lengths of the sequences in each bin against the bin numbers. Strikingly, the pattern is a virtual mirror image of that depicted in Figure 2, despite the fact that the *y*-axis measures a different unit: proportion of formulas learned in Figure 2 and average sequence length per bin in Figure 3. In the present context, the pattern signifies that, for both children, the average length of very low and very high scoring sequences is very short; however, sequences that were merged on the basis of a mid-range score are, on average, considerably longer.

The isomorphy between the plots in Figure 2 and Figure 3 suggests that perhaps the variable which holds all the predictive power for the proportion of sequences learned is average length, not (normalized) log-likelihood bin. Indeed, in Figure 4, we show average sequence lengths against the proportion of sequences learned for each child, and the apparent correlation between these two suggests that, somewhat

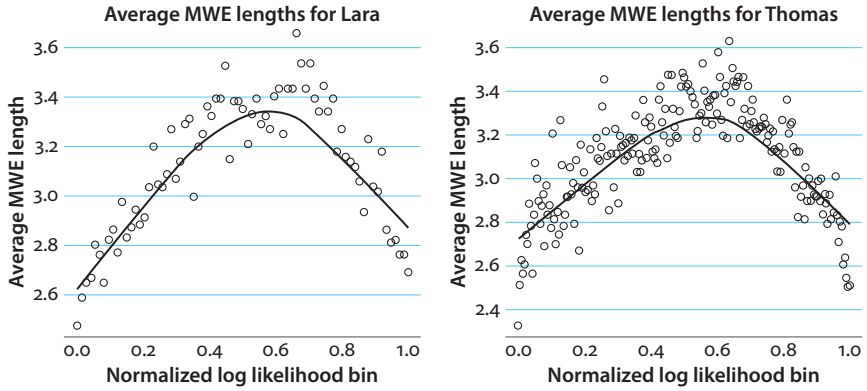

Figure 3. Average sequence lengths as a function of normalized bin rank

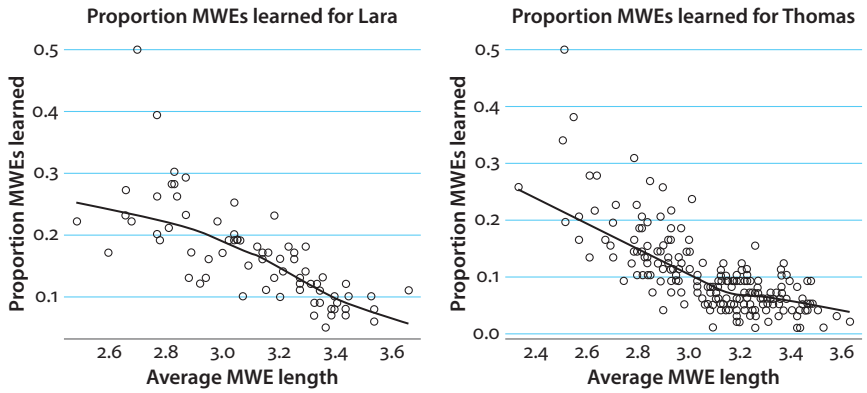

Figure 4. Proportion sequences learned as a function of average sequence lengths

unsurprisingly, average length may be strongly predictive of the dependent variable. This appears particularly true for higher average lengths, where all data points correspond to a low proportion of sequences learned. Note, however, that for shorter average lengths, there are data points which correspond to both rather high and rather low proportions of sequences learned.

To investigate this empirically, we combined the data from the two children and applied a linear model to it. Proportions of sequences learned served as the dependent variable (PROPSrt); to avoid violations of linear model assumptions, we used the square root of the dependent variable (PROPS), while child (CHILD), normalized normalised log-likelihood bin (BIN), and average sequence length (AVELEN) served as predictors. CHILD was a binary variable (Lara vs. Thomas), while all others were numeric. We began with a maximal model in which all numeric predictors were entered as a polynomial to the second degree (to allow for

curvature in the effects) and in which all predictors could interact with each other; model selection tested for the elimination of the polynomial terms and all other predictors. The final model's formula was PROPSrt ~ AVELEN * poly(BIN, 2) * KID (that three-way interaction was very significant: $p = 0.0076$) and that model was highly significant ($F_{11, 276} = 93.54$, $p<10^{-15}$) and achieved a rather high variance explanation (mult. $R^2 = 0.7885$, adj. $R^2 = 0.7801$). All regression coefficients for the model are provided in the appendix, and model checking (homoscedasticity and normality of residuals as well as autocorrelation) raised no red flags.

In Figures 5, 6, and 7, we provide visual representations of the predicted proportions from the final model. Two different perspectives are shown; let us begin with Figure 5 and Figure 6, which are contour plots in which the *x*- and *y*-axes represent the predictors AVELEN and BIN respectively, and the colours and lines displayed represent the predicted proportions of learned formulaic sequences for each combination of the two predictors; for instance, in Figure 5, the plot indicates that the model predicts (and remember that the amount of explained variance was quite high) that, when AVELEN is 3 and BIN is medium (0.5), then the proportion of learned formulas for Lara is about 40% (0.4).

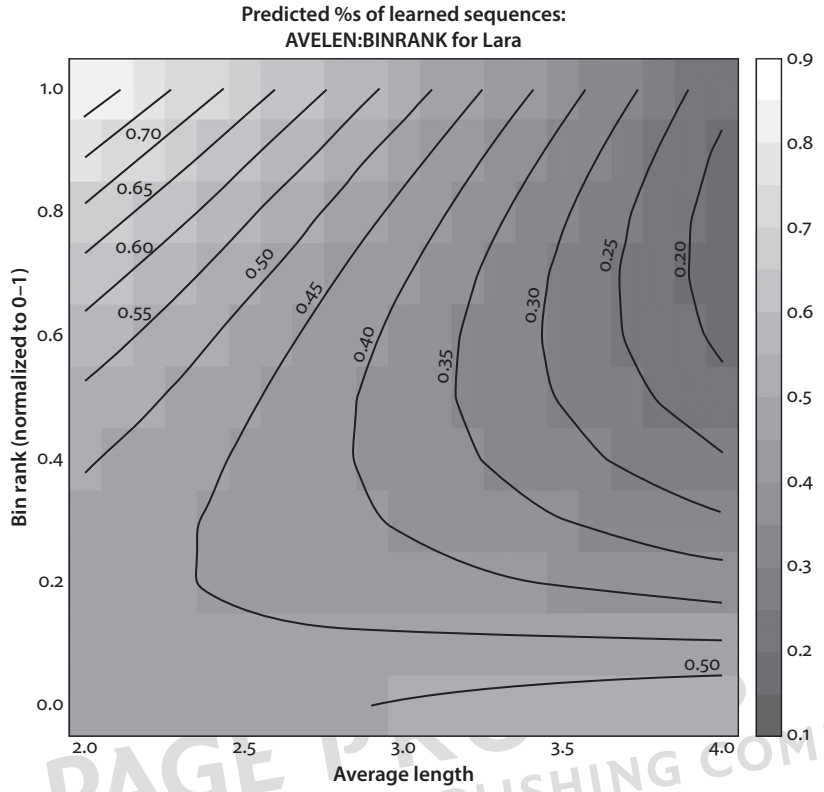

Figure 5. Contour plot of the regression surface of the final model for Lara

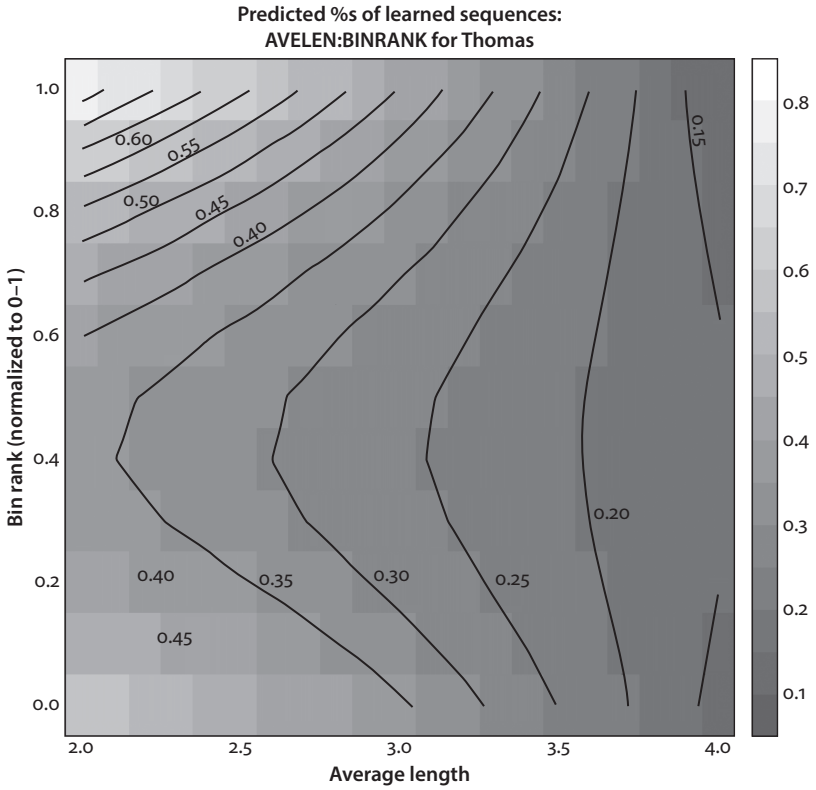

Figure 6. Contour plot of the regression surface of the final model for Thomas

These regression surfaces show that, for both children, short formulas with high log-likelihood scores are learned well/best, and long formulas with moderate to high log-likelihood scores are learned badly. The main difference between the two children is found with low log-likelihood scores: For Lara, there is an effect such that formulas with low log-likelihood scores are learned intermediately well regardless of their length, in fact with a tiny increase for the longer formulas; that finding is not compatible with a long history of research findings on child acquisition and, thus, is somewhat counterintuitive, but it has to be noted that the effect is very small (about a mere 5%) and, for instance, for the BIN-values of 0, 0.1, and 0.2 the slope of the regression surface is statistically not different from 0 (as judged from the predictions' 95%-confidence intervals). For Thomas, the results are more compatible with 'received wisdom': Across all BIN-values, longer formulas are learned less well than shorter ones, but this effect of AVELEN is weakest for intermediately high log-likelihood scores.

With the exception of the (insignificant) slope of the regression surface for low log-likelihood values of Lara, these results make sense and provide some first evidence for higher MERGE bins being learned better even when length is controlled

for. However, it needs to be borne in mind that Figure 5 / Figure 6 provide predictions for all possible combinations of AVELEN and BIN – nevertheless, most of the combinations that are mathematically possible are actually not attested in the data, which makes it useful to consider the predictions specifically for the ranges of combinations of values that *are* attested, which is what is represented in Figure 7. In each panel (one for each child), the *x*- and *y*-axes are the same as in the contour plots above, but now the predicted proportion is represented by an integer value from 0 (lowest predicted proportion) to 9 (highest predicted proportion). In other words, the integer values within the plots can be thought of as 'relative elevations' corresponding to the different predicted proportions of sequences learned, given the intersecting values of the two predictors. In addition to the number, the physical font size of the plotted number represents the predicted proportion as an additional visual clue.

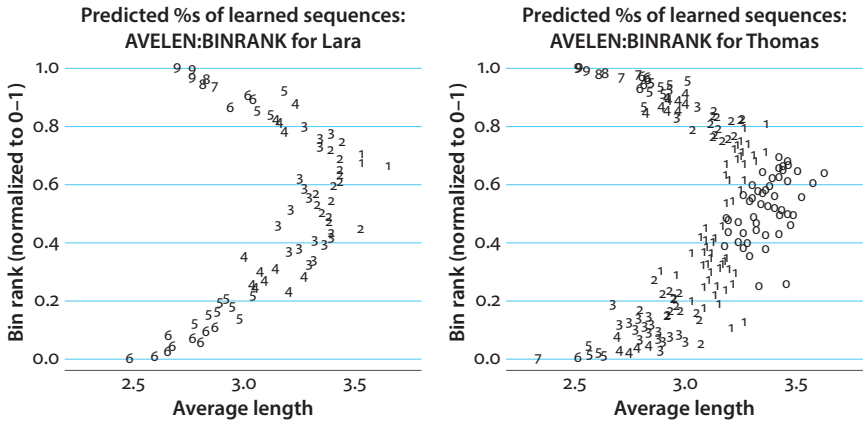

Figure 7. Regression surface of the final model for Lara (left panel) and Thomas (right panel)

This visualization of predicted values – now for those combinations of AVELEN and BIN that are attested – makes the trends even clearer: it is very apparent that there is an overall effect of AVELEN such that both children learn formulas worse the longer they are. For Lara, this effect is weaker, while for Thomas it is stronger. At the same time, one can just as clearly see that, for any observed formula length, the formulas with high BIN-values are learned better. Consider for instance the left panel for Lara, namely when AVELEN is between 2.75 and 3: in those cases, when BIN is low, the predicted values are represented with values ranging from 6 to 4, but when BIN is high, the predicted values range from 9 to 6 respectively. A similar case can be made for Thomas, if one considers AVELEN-values between 2.5 and 3.1: for every predicted value when BIN is low, the corresponding values

for when BIN is high are (sometimes considerably) higher – 6 to 1 compared to 9 to 2. In other words, and as anticipated, when BIN (i.e. MERGE values) are higher, the children learn the formulaic sequences better.

4.3 Discussion

To summarize, these children are averse to learning long sequences, regardless of the association strengths. Given that they are in the age range of 2–3, this is unsurprising, since longer multiword utterances are rare in the speech of children of this age. However, association strength indeed has an effect for all but the longest average lengths: as expected, in the case of both children, higher-strength sequences (as registered by their BIN rank) are learned at a higher rate than lower-strength sequences. In the future, it would be desirable to use longitudinal corpora from slightly older children who produce longer, more complex utterances to determine whether the same effect for short *n*-grams observed here may be likewise observed for longer *n*-grams.

More generally, we have shown that the MERGE algorithm can indeed be methodologically deployed in a theoretical application that studies formulaic language. Whether, in this particular application, automatically extracted formulaic sequences would perform better than manually annotated ones is an open question (and not one that we set out to address in this case study). However, we wish to point out that it is not clear in the first place that the formulaic sequences that children detect in caregiver input and in turn use to bootstrap their own language production are necessarily the same ones that adult annotators would identify as true formulaic sequences. Rather, it may be that the bottom-up approach of automatic extraction in general and lexical association in particular, while obviously imperfect, exhibit closer parallels to the frequency-based acquisition mechanisms employed by children than do whatever crystallized lexical knowledge that adult annotators use.

5. Conclusion

Formulaic language has become a major focus of research in linguistics, as scholars have realized how fundamental and omnipresent it is in discourse. Accordingly, techniques for its efficient identification in textual data are much in demand. While manual annotation is still considered the technique that offers the highest precision, the degree of recall it can offer is more limited given its high costs and time requirements (esp. once interrater reliability is also considered), which has led to great interest in the development of effective computational extraction algorithms. Many of the existing algorithms exhibit shortcomings, though, including the use of

statistical measures for scoring candidate sequences that are either (1) limited to bigrams, or (2) insensitive to high-frequency, low-contingency sequences. Moreover, basic approaches tend to extract many partially redundant, overlapping, and fragmentary sequences.

In this paper, we have presented and tested an algorithm that addresses various issues. Entitled MERGE (for Multi-word Expressions from the Recursive Grouping of Elements), the algorithm employs a *recursive bigram approach*, whereby it is able to grow formulaic sequences of any length in a bottom-up fashion, all while never having to calculate statistical associations for anything other than simple 2-way co-occurrences. As we have shown, MERGE stands up well against another extraction algorithm from the literature, the Adjusted Frequency List, when compared to manually annotated formulaic sequences from the British National Corpus (BNC). What is more, we have shown that MERGE can be successfully used to help predict word sequences that young children learn based on their caregiver input, lending support to the idea that automatic extraction algorithms are viable methodological tools for application in formulaic language research. But despite these successes, it is clear from case study 1 that MERGE still neglects to identify many formulaic sequences identified by the BNC annotators. Thus, further refinement of automatic techniques such as MERGE is still needed.

Along these lines of further refinement, MERGE allows for the identification of formulaic sequences that may contain one or more gaps of various sizes. However, in the present case studies, this ability was not exploited/tested. In the future, it would be desirable to investigate what benefits, if any, this built-in capacity yields. Does it improve the performance of the identification of continuous sequences by offering more paths to a particular formulaic sequence (*in spite + of* versus *in + spite of + in _ of + spite*)? Does it indeed result in the identification of true discontinuous formulaic sequences or does it not result in performance gains?

Note that paradigmatic slots within formulaic sequences (and at their edges for that matter) may be filled with constituents of different lengths in words (e.g., *as small as* versus *as vanishingly small as*). However, as it is currently implemented, MERGE would not treat, say, *as _ as* and *as _ _ as* as the same type, even though they clearly are. Again, further development of the algorithm is needed, given that formulaic sequences comprise not only frozen lexical items but they also allow for different kinds of – and varying degrees of – schematicity (see Langacker, 1987; Goldberg, 1995; 2006; or Bybee, 2010 for discussion of the many different levels of schematicity/generality of the mental lexicon/constructicon), which in turn suggests that, down the road, using an association measure, or a combination of measures that also incorporate type frequencies or type entropies, might be useful. Currently, however, we submit that MERGE offers a state-of-the-art approach to the automatic identification of formulaic sequences.

References

Altenberg, B. (1998). On the phraseology of spoken English: The evidence of recurrent word-combinations. In A. P. Cowie (Ed.), *Phraseology: Theory, analysis, and applications* (pp. 101–102). Oxford: Oxford University Press.

Arnon, I., & Snider, N. (2010). More than words: Frequency effects for multi-word phrases. *Journal of Memory and Language* 6, 67–82. https://doi.org/10.1016/j.jml.2009.09.005

Bannard, C., Lieven, E., & Tomasello, M. (2009). Modeling children's early grammatical knowledge. *Proceedings of the National Academy of Science* 106(41), 17284–17289. https://doi.org/10.1073/pnas.0905638106

Biber, D., Conrad, S., & Cortes, V., (2004). *If you look at …*: Lexical bundles in university teaching and textbooks. *Applied Linguistics*, 25(3), 371–405. https://doi.org/10.1093/applin/25.3.371

Bod, R. (2009). From exemplar to grammar: A probabilistic analogy-based model of language learning. *Cognitive Science*, 33(5), 752–793. https://doi.org/10.1111/j.1551-6709.2009.01031.x

Bolinger, D. (1976). Meaning and memory. *Forum Linguisticum* 1, 1–14.

Bybee, J. (2010). *Language, usage, and cognition*. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511750526

Daudaravičius, V., & Marcinkevičienė, R. (2004). Gravity counts for the boundaries of collocations. *International Journal of Corpus Linguistics*, 9(2), 321–348. https://doi.org/10.1075/ijcl.9.2.08dau

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1), 61–74.

Ellis, N. C., Simpson-Vlach, R., & Maynard, C. (2008). Formulaic language in native and second language speakers: Psycholinguistics, corpus linguistics, and TESOL. *TESOL Quarterly*, 42(3), 375–396. https://doi.org/10.1002/j.1545-7249.2008.tb00137.x

Erman, B., & Warren, B. (2000). The idiom principle and the open choice principle. *Text*, 20(1), 29–62. https://doi.org/10.1515/text.1.2000.20.1.29

Evert, S. (2004). *The statistics of word co-occurrences: Word pairs and collocations*. (PhD Thesis, Universität Stuttgart).

Evert, S. (2009). Corpora and collocations. In A. Lüdeling, & M. Kytö (Eds.), *Corpus linguistics: an international handbook*, Vol. 2 (pp. 1212–1248). Berlin/New York: Mouton de Gruyter.

Goldberg, A. E. (1995). *Constructions: a construction grammar approach to argument structure*. Chicago: University of Chicago Press.

Goldberg, A. E. (2006). *Constructions at work*. Oxford: Oxford University Press.

Granger, S., & Meunier, F. (Eds). (2008). *Phraseology. An interdisciplinary perspective*. Amsterdam/Philadelphia: John Benjamins. https://doi.org/10.1075/z.139

Gries, S. Th. (2008). Dispersions and adjusted frequencies in corpora. *International Journal of Corpus Linguistics*, 13(4), 403–437. https://doi.org/10.1075/ijcl.13.4.02gri

Gries, S. Th., & Mukherjee, J. (2010). Lexical gravity across varieties of English: An ICE-based study of *n*-grams in Asian Englishes. *International Journal of Corpus Linguistics*, 15(4), 520–548. https://doi.org/10.1075/ijcl.15.4.04gri

Gries, S. Th. (2015). Some current quantitative problems in corpus linguistics and a sketch of some solutions. *Language and Linguistics*, 16(1), 93–117. https://doi.org/10.1177/1606822X1455660

Langacker, R. W. (1987). *Foundations of cognitive grammar: Vol. 1: Theoretical prerequisites*. Stanford: Stanford University Press.

Lieven, E., Salomo D., & Tomasello, M. (2009). Two-year-old children's production of multiword utterances: A usage-based analysis. *Cognitive Linguistics*, 20(3), 481–507. https://doi.org/10.1515/COGL.2009.022

MacWhinney, B. (2000). *The CHILDES project. Tools for analyzing talk*. Third edition. Mahwah, NJ: Lawrence Erlbaum Associates.

McEnery, T. (2006). *Swearing in English: Bad language, purity and power from 1586 to the present*. Abington: Routledge.

O'Donnell, M. B. (2011). The adjusted frequency list: A method to produce cluster-sensitive frequency lists. *ICAME Journal*, 35, 135–169.

Pecina, P. (2009). *Lexical association measures: Collocation extraction*. Prague: Charles University.

Rowland, C. F., & Fletcher, S. L. (2006). The effect of sampling on estimates of lexical specificity and error rates. *The Journal of Child Language* 33(4), 859–877. https://doi.org/10.1017/S0305000906007537

Swingley, D. (2005). Statistical clustering and the contents of the infant vocabulary. *Cognitive Psychology*, 50, 86–132. https://doi.org/10.1016/j.cogpsych.2004.06.001

Tomasello, M. (2005). *Constructing a language: a usage-based theory of language acquisition*. Cambridge, MA: Harvard University Press.

Wahl, A. (2015). Intonation unit boundaries and the storage of bigrams: Evidence from bidirectional and directional association measures. *Review of Cognitive Linguistics*, 13(1), 191–219. https://doi.org/10.1075/rcl.13.1.08wah

Wahl, A., & Gries, S. Th. (2018). Multi-word expressions: A novel computational approach to their bottom-up statistical extraction. In P. L. Cantos-Gómez and M. Almela-Sánchez (Eds.), *Lexical collocation analysis: advances and applications* (pp. 85–109). Berlin/New York: Springer

Wible, D., Kuo, C., Chen, M., Tsao, N., & Hung, T. (2006). A computational approach to the discovery and representation of lexical chunks. *TALN* 2006. Leuven, Belgium.

Wray, A. (2002). *Formulaic language and the lexicon*. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511519772

Appendix. Summary statistics for the linear model on the acquisition data

Predictor	*b*	*se*	*t*	$p_{\text{two-tailed}}$
Intercept	0.87926	0.18777	4.683	<0.001
AVELEN	−0.15195	0.06040	−2.516	0.012445
BIN	5.29464	1.34825	3.927	<0.001
poly(BIN, 2)	0.26540	1.29413	0.205	0.837663
CHILD $_{\text{Lara → Thomas}}$	−0.08024	0.20476	−0.392	0.695463
AVELEN : BIN	−1.72888	0.47150	−3.667	<0.001
AVELEN : poly(BIN, 2)	0.14818	0.43396	0.341	0.733017
AVELEN: CHILD $_{\text{Lara →Thomas}}$	−0.01156	0.06589	−0.175	0.860909
BIN : CHILD $_{\text{Lara →Thomas}}$	−3.26269	1.54239	−2.115	0.035296
poly(BIN, 2) : CHILD $_{\text{Lara →Thomas}}$	3.03135	1.48600	2.040	0.042309
AVELEN : BIN : CHILD $_{\text{Lara →Thomas}}$	1.19420	0.53703	2.224	0.026976
AVELEN : poly(BIN, 2) : CHILD $_{\text{Lara →Thomas}}$	−1.00916	0.49722	−2.030	0.043358