Unsupervised Multiword Segmentation of Large Corpora using Prediction-Driven Decomposition of *n*-grams

Julian Brooke[*†] **Vivian Tsang**[†] **Graeme Hirst**[*] **Fraser Shein**[*†]

[*]Department of Computer Science
University of Toronto
jbrooke@cs.toronto.edu
gh@cs.toronto.edu

[†]Quillsoft Ltd.
Toronto, Canada
vtsang@quillsoft.ca
fshein@quillsoft.ca

Abstract

We present a new, efficient unsupervised approach to the segmentation of corpora into multiword units. Our method involves initial decomposition of common *n*-grams into segments which maximize within-segment predictability of words, and then further refinement of these segments into a multiword lexicon. Evaluating in four large, distinct corpora, we show that this method creates segments which correspond well to known multiword expressions; our model is particularly strong with regards to longer (3+ word) multiword units, which are often ignored or minimized in relevant work.

1 Introduction

Identification of multiword units in language is an active but increasingly fragmented area of research, a problem which can limit the ability of others to make use of units beyond the level of the word as input to other applications. General research on word association metrics (Church and Hanks, 1990; Smadja, 1993; Schone and Jurafsky, 2001; Evert, 2004; Pecina, 2010), though increasingly comprehensive in its scope, has mostly failed to identify a single best choice, leading some to argue that the variety of multiword phenomena must be tackled individually. For instance, there is a body of research focusing specifically on collocations that are (to some degree) non-compositional, i.e. multiword expressions (Sag et al., 2002; Baldwin and Kim, 2010), with individual projects often limited to a particular set of syntactic patterns, e.g. verb-noun combinations (Fazly et al., 2009). A major issue with approaches involving statistical association is that they rarely address expressions larger than 2 words (Heid, 2007); in corpus linguistics, larger sequences referred to as *lexical bundles* are extracted using an *n*-gram frequency cutoff (Biber et al., 2004), but the frequency threshold is typically set very high so that only a very limited set is extracted. Another drawback, common to almost all these methods, is that they rarely offer an explicit segmentation of a text into multiword units, which would be preferable for downstream uses such as probabilistic distributional semantics. An exception is the Bayesian approach of Newman et al. (2012), but their method does not scale well (see Section 2). Our own long-term motivation is to identify a wide variety of multiword units for assisting language learning, since correct use of collocations is known to pose a particular challenge to learners (Chen and Baker, 2010).

Here, we present a multiword unit segmenter[1] with the following key features:

- It is entirely unsupervised.
- It offers both segmentation of the input corpus and a lexicon which can be used to segment new corpora.
- It is scalable to very large corpora, and works for a variety of corpora.
- It is language independent.

[1]The software is available at http:/www.cs.toronto.edu/~jbrooke/ngram_decomp_seg.py .

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 753–761, Dublin, Ireland, August 23-29 2014.

- It does not inherently limit possible units with respect to part-of-speech or length.
- It has a bare minimum of parameters, and can be used off-the-shelf: in particular, it does not require the choice of an arbitrary cutoff for some uninterpretable statistical metric.
- It does, however, include a parameter fixing the minimum number of times that a valid multiword unit will appear in the corpus, which ensures sufficient usage examples for relevant applications.

Our method involves three major steps: extraction of common *n*-grams, initial segmentation of the corpus, and a refinement of the resulting lexicon (and, by extension, the initial segmentation). The latter two steps are carried out using a simple but novel heuristic based on maximizing word prediction within multiword segments. Importantly, our method requires just a few iterations through the corpus, and in practice these iterations can be parallelized. Evaluating with an existing set of multiword units from WordNet in four large corpora from distinct genres, we show that our initial segmentation offers extremely good subsumption of known collocations, and after lexicon refinement the model offers a good trade-off between subsumption and exact matches. We also evaluate a sample of our multiword vocabulary using crowdsourcing, and offer a qualitative analysis.

2 Related Work

In computational linguistics, there is a large body of research that proposes and/or evaluates lexical association measures for the creation of multiword lexicons (Church and Hanks, 1990; Smadja, 1993; Schone and Jurafsky, 2001; Evert, 2004): there are many more measures than can be addressed here—work by Pecina (2010) considered 82 variations—but popular choices include the *t*-test, log likelihood, and pointwise mutual information (PMI). In order to build lexicons using these methods, particular syntactic patterns and thresholds for the metrics are typically chosen. Many of the statistical metrics do not generalize at all beyond two words, but PMI (Church and Hanks, 1990), the log ratio of the joint probability to the product of the marginal probabilities, is a prominent exception. Other measures specifically designed to address collocations of larger than two words include the c-value (Frantzi et al., 2000), a metric designed for term extraction which weights term frequency by the log length of the *n*-gram while penalizing *n*-grams that appear in frequent larger ones, and mutual expectation (Dias et al., 1999), which produces a normalized statistic that reflects how much a candidate phrase resists the omission of any particular word. Another approach is to simply to combine known $n-1$ collocations to form *n*-length collocations (Seretan, 2011), but this is based on the assumption that all longer collocations are built up from shorter ones—idioms, for instance, do not usually work in that way.

An approach used in corpus linguistics which does handle naturally longer sequences is the study of *lexical bundles* (Biber et al., 2004), which are simply *n*-grams that occur above a certain frequency threshold. This includes larger phrasal chunks that would be missed by traditional collocation extraction, and so research in this area has tended to focus on how particular phrases (e.g. *if you look at*) are indiciative of particular genres (e.g. university lectures). In order to get very reliable phrases, the threshold is typically set high enough (Biber et al. use 40 occurrences in 1 million words) to filter out the vast majority of expressions in the process.

With respect to the features of our model, the work closest to ours is probably that of Newman et al. (2012). Like us, they offer an unsupervised solution, in their case a generative Dirichlet Process model which jointly creates a segmentation of the corpus and a multiword term vocabulary. Their method, however, requires full Gibbs sampling with thousands of iterations through the corpus (Newman et al. report using 5000), an approach which is simply not tractable for the large corpora that we address in this paper (which are roughly 1000 times larger than theirs). Though the model is general, their focus is limited to term extraction, and for larger terms they compare only with the c-value approach of Frantzi et al. (2000). Other closely related work includes general tools available for creating multiword lexicons using association measures or otherwise exploring the collocational behavior of words (Kilgarriff and Tugwell, 2001; Araujo et al., 2011; Kulkarni and Finlayson, 2011; Pedersen et al., 2011). Other related but distinct tasks include syntactic chunking (Abney, 1991) and word segmentation for Asian languages, in particular Chinese (Emerson, 2005).

3 Method

3.1 Prediction-based segmentation

Our full method consists of multiple independent steps, but it is based on one central and relatively simple idea that we will introduce first. Given a sequence of words, $w_1 \ldots w_n$, and statistics (i.e. n-gram counts) about the use of these words in a corpus, we first define $p(w_i|w_{j,k})$ as the conditional probability of some word w_i appearing with some contextual subsequence $w_j \ldots w_{i-1}, w_{i+1} \ldots w_k, 1 \leq j \leq i \leq k \leq n$. In the case $i = j = k$, this is simply the marginal probability, $p(w_i)$. We then define the word predictability of some w_i in the context $w_{1,n}$ as the log of the maximal conditional probability of the word across all possible choices of j and k:

$$pred(w_i, w_{1,n}) = \max_{j,k} \log p(w_i|w_{j,k})$$

We can define predictability for the entire sequence then as:

$$pred(w_{1,n}) = \sum_{i=1}^{n} pred(w_i, w_{1,n})$$

Now we consider the case where we have a set of possible segmentations S of the sequence, where each segmentation $s \in S$ can be viewed as a (possibly empty) set of segment boundaries $\langle s_0, s_1, \ldots, s_m \rangle$. Among the available options, our optimal segmentation is:

$$\arg\max_{s \in S} \sum_{i=0}^{m-1} pred(w_{s_i, s_{i+1}-1})$$

That is, we will prefer the segmentation which maximizes the overall predictability of each word in the sequence, under the restriction that we only predict words using the context within their segments. This reflects our basic assumption that words within a good segment, i.e. a multiword unit, are (much) more predictive of each other than words outside a unit. Note that if our probabilities are calculated from the full set of n-gram counts for the corpus being segmented and the set of possible segmentations S is not constrained, a segmentation with a smaller number of breaks will generally be preferred over one with more breaks. However, in practice we will be greatly constraining S and also using probabilities based on only a subset of all the information in the corpus.

3.2 Extraction of n-grams

In order to carry out a segmentation of the corpus using this method, we first need to extract statistics in the form of n-gram counts. Given a minimum occurrence threshold, this can be done efficiently even for large corpora in an iterative fashion until all n-grams have been extracted. For all our experiments here, we limit ourselves to n-grams that appear at least once in 10 million tokens, and we did not collect n-grams for $n > 10$ (which are almost always the result of duplication of texts in the corpus). For the purposes of calculating conditional probabilities given surrounding context in our predictive segmentation, we collected both standard n-grams as well as (for $n \geq 3$) skip n-grams with a missing word (e.g. *basic * processes* where the asterisk indicates that any word could appear in that slot). Here we use lower-cased unlemmatized tokens, excluding punctuation, though for languages with more inflectional morphology than English, lemmatization would be advised.

3.3 Initial segmentation

Given these n-gram statistics, our initial segmentation proceeds as follows: For each sentence in the corpus, we identify all maximum length n-grams in the sentence, i.e. all those n-grams for $n \geq 2$ where there is no larger n-gram which contains them while still being above our threshold of occurrence. These n-grams represent the upper bound of our segmentation: we will never break into segments larger than these. However, there are many overlaps among these n-grams (in fact, with a low threshold the vast majority of n-grams overlap with at least one other), and for proper segmentation we need to resolve

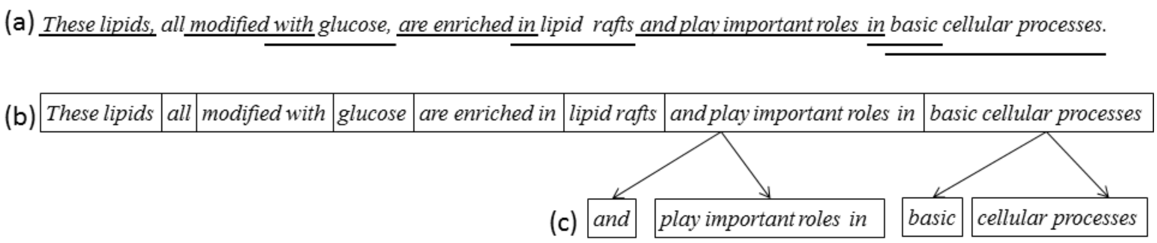

Figure 1: Three-step procedure for n-gram decomposition into multiword units. a) shows the maximal n-grams identified in the sentence, b) is the segmentation after the initial pass of the corpora, and c) shows further decomposition of segments after a pass through the lexicon resulting from b).

all overlaps between these maximal n-grams by inserting at least one break. For this we apply our prediction-based decomposition technique. In our discussion in Section 3.1, we did not consider how the possible segmentations were selected, but now we can be explicit: the set S consists of all possible segmentations which minimally resolve all n-gram overlaps. By minimally resolve, we mean that the removal of any breakpoint from our set would result in an unresolved overlap: in short, there are no extraneous breaks, and therefore no cases where a possible set of breaks is a subset of another possible set. Figure 1a shows a real example: if we just consider the last three maximal n-grams, there are two possible minimal breaks: a single break between *in* and *basic* or two breaks, one between *roles* and *in* and one between *basic* and *cellular*.

Rather than optimizing over all possible breaks over the whole sentence, which is computationally problematic, we simplify the algorithm somewhat by moving sequentially through each n-gram overlap in the sentence, taking any previous breaks as given while considering only the minimum breaks necessary to resolve any overlaps that directly influence the segmentation of the two overlapping spans under consideration, which is to say any other overlapping spans which contain at least one word also contained in at least one of overlapping spans under consideration. For example, in Figure 1a we first deal independently with each of the first two overlaps (the spans *modified with glucose* and *are enriched in lipid rafts*, and then we consider the final two overlaps together: The result is shown in Figure 1b. In development, we tested including more context (i.e. considering second-order influence) and found no benefit. Since we do not consider breaks other than those required to resolve overlapping n-grams, these segments tend to be long. This is by design; our intention is that these segments will subsume as many multiword units as possible, and therefore will be amenable to refinement by further decomposition in the next step.

3.4 Lexicon decomposition

Based on the initial segmentation of the entire corpus, we extract a tentative lexicon, with corresponding counts. Then, in order from longest to shortest, we consider decomposition of each entry. First, using our prediction-based decomposition method, we find the best decomposition of the entry into two parts; note that we only need to consider one break per lexicon entry, since breaks in the (smaller) parts will be considered independently later in the process. If the count in our lexicon is below the occurrence threshold, we always carry out this split, which means we remove the entry from the lexicon and (after all n-grams of that length have been processed, so as to avoid ordering effects) add its counts to the counts of n-grams of its best decomposition. If the count is above the threshold, we preserve the full entry (for entries of length 3 or greater) only if the following inequality is true for each subsegment $w_{j,k}$ in the full entry $w_{1,n}$:

$$\sum_{i=j}^{k} pred(w_i, w_{1,n}) - pred(w_{j,k}) > \log p(w_{j,k}) - \log p(w_{1,n})$$

That is, the ratio (expressed as a difference of logarithms) between the count of the segment and the full unsegmented entry (in our preliminary lexicon) is lower than the ratio of the predictability (as defined in our discussion of prediction-based decomposition) of the words in the segment with the context of the full entry to the predictability of words with only the context included within the segment (which is just $pred(w_{j,k})$). In other words, we preserve only longer multiword sequences in our lexicon when any decrease in the probability of the full entry relative to its smaller components[2] is fully offset by an increase in the conditional probability of the individual words of that segment when the larger context from the full segment is available. For example, after we have decided on a potential break in the phrase *basic* | *cellular process* from our example in Figure 1, we compare the (marginal) probability "lost" by including *basic* in a larger phrase, i.e. the ratio of counts of *basic* to *basic cellular process* in our lexicon), to the (conditional) probability "gained" by how much more predictable the segment is in this context; when the segment in question is a single word, as in this case, this is simply $p(basic|cellular\ process)/p(basic)$, and we break only when there is more gain than loss. This restriction could be parameterized for more fine-grained control of the trade-off between larger and smaller segments in specific cases, but in the interest of avoiding nuisance parameters we just use it directly. Once we have decomposed all eligible entries to create a final lexicon, we apply these same decompositions to the segments in our initial segmentation to produce a final segmentation (see Figure 1c).

4 Evaluation

Multiword lexicons are typically evaluated in one of two ways: direct comparison to an existing lexicon, or precision of the top n candidates offered by the model. There are problems with both these methods, since there are no resources that offer a truly comprehensive treatment of multiword units, defined broadly, and the top n candidates from a model for small n may not be a particularly representative sample: in particular, they might not include more common terms, which should be given more weight when one is considering downstream applications. Given the dual output of our model, evaluation using segmentation is another option, except that creating full gold standard segmentations would be a particularly difficult annotation task, since our notion of multiword unit is a broad one.

In light of this, we evaluate by taking the best from these various approaches. Given an existing multiword lexicon, we can evaluate not by comparing our lexicon to it directly, but rather by looking at the extent to which our segmentation preserves these known multiword units. There are several major advantages to this approach: first, it does not require a full lexicon or gold standard segmentation; second, common units are automatically given more weight in the evaluation; third, we can use it for evaluation in very large corpora. Our two main metrics are *subsumption* (Sub), namely the percentage of multiword tokens that are fully contained with a segment, and *exact matches* (Exact), the percentage of multiword tokens which correspond exactly to a segment. Exact matches would seem to be preferable to subsumption, but in practice this is not necessarily the case, since our method often identifies valid compound terms and larger constructions than our reference lexicon contains; for example, WordNet only contains the expression *a lot*, but when appearing as part of a noun phrase our model typically segments this to *a lot of*, which, in our opinion, is a preferable segmentation. To quantify overall performance, we calculate a harmonic mean (Mean) of the two metrics. We also looked specifically at performance for terms of 3 or more words (Mean 3+), which are less studied and more relevant to our interests.

Our second evaluation focuses on the quality of these longer terms with a post hoc annotation of output from our model and the best alternatives. We randomly extracted pairs of segments of three words or more where our model mostly but not entirely overlapped with an alternative model (750 examples per corpus per method), and asked CrowdFlower workers to choose which output seemed to be a better multiword unit in the context; they were shown the entire sentence with the relevant span underlined, and then the two individual chunks separately. To ensure quality, we used our multiword lexicon to

[2] This probability is based on the respective counts in our preliminary lexicon at this step in the process, not the original n-gram probability. One key advantage to doing the initial segmentation first is that words that appear consistently in larger units, an extreme example is the bigram *vector machine* in the term *support vector machine*, already have low or zero probability, and will not appear in the lexicon or be good candidate segments for decomposition. This rather intuitively accomplishes what the c-value metric is modeling by applying negative weights to candidates appearing in larger n-grams.

create gold standard examples (comparing known multiword units to purposely bad segmentations which overlapped with them), and used them to test and filter out unreliable workers: for inclusion in our final set, we required a minimum 90% performance on the test questions. We also limited each contributor to only 250 judgments, so that our results reflected a variety of opinions.

We considered a number of alternatives to our approach, though we limited the comparison to methods which could predict segments greater than 2 words, those that were computationally feasible for large corpora, and those which segment into single words only as a last resort: approaches which prefer single words cannot do well under our evaluation because we have no negative examples, only positive ones. The majority of our alternatives involve ranking all potential n-grams (not just the maximal) with $n \geq 2$ and then greedily segmenting them: big-n prefers longer n-grams (with a backoff to counts); c-value is used for term extraction (Frantzi et al., 2000) and was also compared to by Newman et al. (2012); ME refers to the Mutual Expectation metric (Dias et al., 1999); and PMI uses a standard extension of PMI to more than 2 words. We also tested standard (pairwise) PMI as a metric for recursively joining contiguous units (starting with words) into larger units until no larger units can be formed (PMI join), and a version of our decomposition algorithm which selects the minimal breaks which maximize total word count across segments rather than total word predictability (count decomp); the fact that traditional association metrics are not defined for single words prevents us from using them as alternatives to predictability in our decomposition approach. Finally, we also include an oracle which chooses the correct n-grams when they are available for segmentation, but which still fails for units that are below our threshold.

We evaluated our model in four large English corpora: news articles from the Gigaword corpus (Graff and Cieri, 2003) (4.9 billion tokens), out-of-copyright texts from the Gutenberg Project[3] (1.7 billion tokens), a collection of abstracts from PubMed (2.2 billion tokens)[4], and blogs from the ICWSM 2009 social media corpus (Burton et al., 2009) (1.1 million tokens). Our main comparison lexicon is WordNet 3.0, which contains a good variety of expressions appropriate to the various genres, but we also included multiword terms from the Specialist Lexicon[5] for better coverage of the biomedical domain. One issue with our evaluation is that it assumes all tokens are true instances of the multiword unit in question; we carried out a manual inspection of multiword tokens identified by string match in our development sets (5000 sentences set aside from each of the abstract and blog corpora), and excluded from the evaluation a small set of idiomatic expressions (e.g. *on it*, *do in*) whose literal, non-MWE usage is too common for the expression to be used reliably for evaluation; otherwise, we were satisfied that the vast majority of multiword tokens were true matches. When one multiword token appeared within another, we ignored the smaller of the two; when two overlapped in the text, we ignored both.

5 Results

All the results for the main evaluation are shown in Table 1. First, we observe that our initial segmentation always provides the highest subsumption, and our final lexicon always provides the highest harmonic mean, with a modest drop in subsumption but a huge increase in exact matches. The alternative models fall roughly into two categories: those which have reasonably high subsumption, but few exact matches (PMI rank seems to be the best of these) and those that have many exact matches (sometimes better than either of our models) but are almost completely ineffective for identifying multiword units of length greater than 2 (ME rank and c-value, with ME offering more exact matches): the latter phenomenon is attributable to the predominance of two-word multiword tokens in our evaluation, which means a model can do reasonably well by guessing mostly two-word units. For the corpora with more multiword units of greater length, i.e. the PubMed abstracts and the Gutenberg corpus, our method also provides the most exact matches. Our best results come in the PubMed corpus, probably because the texts are the most uniform, though results are satisfactory in all four corpora tested here, which represent a considerable range of genres.

[3]http://www.gutenberg.org . Here we use the English texts from the 2010 image, with headers and footers filtered out using some simple heuristics.

[4]http://www.ncbi.nlm.nih.gov/pubmed/

[5]http://www.nlm.nih.gov/research/umls/new_users/online_learning/LEX_001.htm

Table 1: Performance in segmenting multiword units of various segmentation methods in 4 large corpora. Sub. = Subsumption (%); Exact = Exact Match (%); Mean = Harmonic mean of Sub and Exact; Mean 3+ = Harmonic mean of Sub and Exact for multiword tokens of at length 3 or more. Bold is best in column for corpus, excluding the oracle.

Method	Gigaword news articles				Gutenberg texts			
	Sub	**Exact**	**Mean**	**Mean 3+**	**Sub**	**Exact**	**Mean**	**Mean 3+**
Oracle	97.1	97.1	97.1	95.5	97.0	97.0	97.0	97.8
big-*n* rank	88.7	28.8	43.5	51.4	84.9	30.1	44.4	57.5
c-value rank	69.1	66.1	67.6	23.3	58.6	57.7	58.2	12.6
ME rank	75.3	**70.0**	72.6	14.4	63.2	61.0	62.1	10.9
PMI rank	90.8	30.0	45.1	53.5	86.9	32.8	47.7	61.2
PMI join	83.1	32.8	47.0	43.7	77.7	32.6	46.0	45.5
Count decomp	75.9	31.3	44.3	47.1	69.2	31.5	43.3	54.2
Prediction decomp, initial	**92.2**	36.4	52.2	**64.4**	**89.3**	38.7	54.0	**71.6**
Prediction decomp, final	85.6	66.4	**75.2**	63.8	78.9	**62.8**	**70.0**	61.6

Method	PubMed abstracts				ICWSM blogs			
	Sub	**Exact**	**Mean**	**Mean 3+**	**Sub**	**Exact**	**Mean**	**Mean 3+**
Oracle	91.9	91.9	91.9	84.0	96.5	96.5	96.5	99.4
big-*n* rank	82.2	40.1	53.9	55.5	86.1	33.3	48.0	60.8
c-value rank	63.2	62.3	62.7	21.7	64.3	62.4	63.3	14.6
ME rank	68.5	65.8	67.1	9.1	69.7	**66.2**	67.9	11.7
PMI rank	87.0	41.4	56.1	58.3	88.4	35.7	50.8	63.4
PMI join	79.8	39.7	53.0	46.8	80.3	35.4	49.1	47.0
Count decomp	71.0	38.4	49.9	50.4	71.5	33.5	45.6	53.9
Prediction decomp, initial	**88.6**	50.3	64.1	67.2	**90.5**	40.3	55.8	**70.9**
Prediction decomp, final	85.2	**73.4**	**78.8**	**69.5**	83.2	64.9	**72.9**	66.9

Table 2: CrowdFlower pairwise preference evaluation, our full model versus a selection of alternatives

Comparison	Preference for Prediction decomp, final
Prediction decomp, final vs. ME	57.9%
Prediction decomp, final vs. Multi PMI	71.0%
Prediction decomp, final vs. Prediction decomp, initial	70.5%

For our crowdsourced evaluation, we compared our final model to the best models of each of the two major types from the first round, namely Mutual Expectation and PMI rank, as well as our initial segmentation. The results are given in Table 2. Our full model is consistently preferred over the alternatives. This is not surprising in the case of the high-subsumption, low-accuracy models, since the resulting segments often have extraneous words included: an example is *in spite of my*, which our model correctly segmented to just *in spite of*. Given that the ME ranking rarely produces units larger than 2 words, however, we might have predicted that when it does it would be more precise than our model, but in fact our model was somewhat preferred (a chi-square test confirmed that this result was statistically different from chance, $p < 0.001$). An example of an instance where our model offered a better segmentation is *call for an end to* as compared to *for an end to* from the ME model, though there are also many instances where the ME segmentation is more sensible, e.g. *what difference does it make* as compared to *difference does it make* from our model.

Looking closer at the output and vocabulary of our model across the various genres, we see a wide range of multiword phenomena: in the medical abstracts, for instance, there is a lot of medical jargon (e.g. *daily caloric intake*) but also other larger connective phrases and formulaic language (e.g. *an alternative explanation for*, *readily distinguished from*). The blogs also have (very different) formulaic language of the sort studied using lexical bundles (e.g. *all I can say is that*, *where else can you*) and lots of idiomatic language (e.g. *reinventing the wheel*, *look on the bright side*). The idioms from the Gutenberg, not surprisingly, tend to be less clichéd and more evocative (e.g. *ghost of a smile*); there are rather stodgy expressions like *far be it from me* and conjunctions we would not see in the other corpora (e.g. *rocks and shoals*, *masters and mistresses*). By contrast, many of the larger expressions in the news articles are from sports and finance (e.g. *investor demand for*, *tied the game with*), with many that would be filtered out using the simple grammatical filters often applied in this space. However, for bigrams in particular, some additional syntactic filtering is clearly warranted.

6 Conclusion

We have presented an efficient but effective method for segmenting a corpus into multiword collocational units, with a particular focus on units of length greater than two. Our evaluation indicates that this method results in high-quality segments that capture a variety of multiword phenomena, and is better in this regard than alternatives based on relevant association measures. This result is consistent across corpora, though we do particularly well with highly stereotyped language such as seen in the biomedical domain.

Future work on improving the model will likely focus on extensions related to syntax, for instance bootstrapped POS filtering and discounting of predictability that can be attributed solely to syntactic patterns. Our method could also be adapted to decompose full syntactic trees rather than sequences of words, offering tractable alternatives to Bayesian approaches that identify recurring tree fragments (Cohn et al., 2009); this would allow us, for instance, to correctly identify constructions with long-distance dependencies or other kinds of variation where relying on the surface form is insufficient (Seretan, 2011).

With regards to applications, we will be investigating how to help learners notice these chunks when reading and then use them appropriately in their own writing; this work will eventually intersect with the well-established areas of grammatical error correction (Leacock et al., 2014) and automated essay scoring (Shermis and Burstein, 2003). As part of this, we will be building distributional lexical representations of these multiword units, which is why our emphasis here was on a highly scalable method. Part of our interest is of course in capturing the semantics of idiomatic phrases, but we note that even in the case when a multiword unit is semantically compositional, it might provide *de facto* word sense disambiguation or be stylistically distinct from its components, i.e. be very specific to a particular genre or sub-genre. Therefore, provided we have enough examples to get reliable distributional statistics, these larger units are likely to provide useful information for various downstream applications.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada and the MITACS Elevate program. Thanks to our reviewers and also Tong Wang and David Jacob for their input.

References

Steven Abney. 1991. Parsing by chunks. In Robert Berwick, Steven Abney, and Carol Tenny, editors, *Principle-Based Parsing*, pages 257–278. Kluwer Academic Publishers.

Vitor De Araujo, Carlos Ramisch, and Aline Villavicencio. 2011. Fast and flexible MWE candidate generation with the mwetoolkit. In *Proceedings of the Multiword Expression Workshop at ACL 2011 (MWE 2011).*

Timothy Baldwin and Su Nam Kim. 2010. Multiword expressions. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group, Boca Raton, FL.

Douglas Biber, Susan Conrad, and Viviana Cortes. 2004. *If you look at...*: Lexical bundles in university teaching and textbooks. *Applied Linguistics*, 25:371–405.

Kevin Burton, Akshay Java, and Ian Soboroff. 2009. The ICWSM 2009 Spinn3r Dataset. In *Proceedings of the Third Annual Conference on Weblogs and Social Media (ICWSM '09)*, San Jose, CA.

Yu-Hua Chen and Paul Baker. 2010. Lexical bundles in L1 and L2 academic writing. *Language Learning & Technology*, 14(2):30–49.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Trevor Cohn, Sharon Goldwater, and Phil Blunsom. 2009. Inducing compact but accurate tree-substitution grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL '09)*.

Gaël Dias, Sylvie Guilloré, and José Gabriel Pereira Lopes. 1999. Language independent automatic acquisition of rigid multiword units from unrestricted text corpora. In *Proceedings of Conférence Traitement Automatique des Langues Naturelles (TALN) 1999*.

Thomas Emerson. 2005. The second international Chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Stefan Evert. 2004. *The statistics of word cooccurences–word pairs and collocatoins*. Ph.D. thesis, University of Stuttgart.

Afsaneh Fazly, Paul Cook, and Suzanne Stevenson. 2009. Unsupervised type and token identification of idiomatic expressions. *Computational Linguistics*, 35(1):61–103.

Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. 2000. Automatic recognition of multi-word terms: the c-value/nc-value method. *International Journal on Digital Libraries*, 3:115–130.

David Graff and Christopher Cieri. 2003. *English Gigaword*. Linguistic Data Consortium, Philadelphia, PA.

Ulrich Heid. 2007. Compuational linguistic aspects of phraseology. In Harald Burger, Dmitrij Dobrovol'skij, Peter Kühn, and Neal R. Norrick, editors, *Phraseology. An international handbook*. Mouton de Gruyter, Berlin.

Adam Kilgarriff and David Tugwell. 2001. Word sketch: Extraction and display of significant collocations for lexicography. In *Proceedings of the ACL Workshop on Collocation: Computational Extraction, Analysis and Exploitation*.

Nidhi Kulkarni and Mark Finlayson. 2011. jMWE: A Java toolkit for detecting multi-word expressions. In *Proceedings of the Multiword Expression Workshop at ACL 2011 (MWE 2011)*.

Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2014. *Automated Grammatical Error Detection for Language Learners (2nd Edition)*. Morgan & Claypool.

David Newman, Nagendra Koilada, Jey Han Lau, and Timothy Baldwin. 2012. Bayesian text segmentation for index term identification and keyphrase extraction. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING '12)*.

Pavel Pecina. 2010. Lexical association measures and collocation extraction. *Language Resources and Evaluation*, 44:137–158.

Ted Pedersen, Satanjeev Banerjee, Bridget McInnes, Saiyam Kohli, Mahesh Joshi, and Ying Liu. 2011. The Ngram statistics package (text::nsp) : A flexible tool for identifying ngrams, collocations, and word associations. In *Proceedings of the Multiword Expression Workshop at ACL 2011 (MWE 2011)*.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLing '02)*.

Patrick Schone and Dan Jurafsky. 2001. Is knowledge-free induction of multiword unit dictionary headwords a solved problem? In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP '01)*.

Violeta Seretan. 2011. *Syntax-Based Collocation Extraction*. Springer.

Mark D. Shermis and Jill Burstein, editors. 2003. *Automated Essay Scoring: A Cross-Disciplinary Approach*. Lawrence Erlbaum Associates, Mahwah, NJ.

Frank Smadja. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, pages 143–177.